CENTRE
DE RECHERCHE
SUR LES TRANSPORTS
(CRT)
CENTRE
FOR RESEARCH
ON TRANSPORTATION

UNIVERSITÉ DE MONTRÉAL
ÉCOLE DES HAUTES ÉTUDES COMMERCIALES
ÉCOLE POLYTECHNIQUE

C.P. 6128, SUCCURSALE A
MONTRÉAL (QUÉBEC)
CANADA
H3C 3J7

TÉLÉPHONE : (514) 343-7575
TÉLEX : 05-24146 BIBPOLYTEC
TÉLÉCOPIEUR (FAX) : (514) 343-7121

CRT-639

STRATEGIC PLANNING OF FREIGHT TRANSPORTATION IN BRAZIL: METHODOLOGY AND APPLICATIONS

Part 2

METHODOLOGY: THE NEW CONTRIBUTIONS

Edited by
Michael Florian and Teodor G. Crainic

This research was supported by the International Development Research Center of Canada, Grant No. 3-P-82-1012 and in part by research grants NSERC (Canada), FCAR (Québec, Canada) and CNPq (Brazil).

Université de Montréal
Centre de recherche sur les transports - Publication # 639
Juillet 1989

90-01418/

Dépôt légal - 3e trimestre 1989

ABSTRACT

This is the final report of the cooperative project ***Strategic Planning of Freight Transportation in Brazil***, carried out jointly by the Centre de Recherche sur les Transports of the Université de Montréal, the Pontifícia Universidade Católica do Rio de Janeiro and GEIPOT, the National Transportation Planning Company of the Brazilian Ministry of Transportation, with the financial support of the International Development Research Centre (Canada). The report consists of five parts:

Part I	Project Summary
Part II	Methodology: The New Contributions
Part III	The STAN System
Part IV	Specifying Data for the STAN System: Requirements, Concepts, Sources and Estimation Procedures
Part V	Applications.

The second part describes the new methodological developments that were motivated by the strategic freight transportation problems considered in the project. It contains the general modeling framework that was developed, as well as the specific representation of various transportation modes that result by using this modeling approach. The formulation and solution algorithms for the multimode multiproduct network assignment method, which is the main new development of the project is described in detail. An approach to computing maximal multimode multiproduct network flows is given as well.

KEY WORDS: Multimode Multiproduct Freight Transportation,
Network Optimization Models,
National/Regional Strategic Analysis and Planning of Freight Flows,
Interactive-Graphic System.

RESUME

Ceci est le rapport final du projet de coopération ***Strategic Planning of Freight Transportation in Brazil***, réalisé conjointement par le Centre de recherche sur les transports de l'Université de Montréal, la Pontifícia Universidade Católica do Rio de Janeiro et GEIPOT, l'agence de planification nationale du ministère des transports du Brésil, avec le support financier du Centre de recherches pour le développement international (Canada). Le rapport a cinq parties:

Part I	Project Summary
Part II	Methodology: The New Contributions
Part III	The STAN System
Part IV	Specifying Data for the STAN System: Requirements, Concepts, Sources and Estimation Procedures
Part V	Applications.

La seconde partie décrit les nouveaux développements méthodologiques réalisés à l'occasion du projet. Nous décrivons tout particulièrement le cadre général de modélisation, ainsi que les représentations des divers modes de transport réalisées à l'aide de cette approche de modélisation. Nous décrivons également, en détail, le modèle et l'algorithme pour la méthode d'affectation multimode multiproduit, qui constitue le principal développement du projet. Une approche pour le calcul des flux multimode multiproduit maximaux est aussi présentée.

MOTS CLES: Transport multimode multiproduit de marchandises,
Modèles d'optimisation de réseaux,
Planification stratégique nationale/régionale,
Système interactif-graphique.

Table of Contents

I **PROJECT SCOPE AND DELIMITATIONS** 1
1.1 PROJECT SCOPE 1
1.2 STRATEGIC PLANNING 1
1.3 METHODOLOGY FOR PREDICTING FLOWS, SERVICES LEVELS AND COSTS 2
1.4 THE DATA BASE 3
1.5 ALTERNATIVE COMPARAISON AND EVALUATION 4

II **SYSTEM MODELING** 5

III **THE MULTI-MODE MULTI-PRODUCT NETWORK ASSIGNMENT MODEL AND ALGORITHM** 17
3.1 THE NETWORK REPRESENTATION 20
3.2 THE MULTI-MODE MULTI-PRODUCT MODEL 24
3.3 A SOLUTION ALGORITHM 28
3.4 CONVERGENCE ANALYSIS OF THE ALGORITHM 32
3.5 THE COST FUNCTIONS 37
3.6 THE SHORTEST PATH ALGORITHM 41
3.7 COMPUTATIONAL RESULTS 44
3.8 CONCLUSIONS 46
3.9 BIBLIOGRAPHY 55

IV **STRATEGIC RAIL MODELING** 57
4.1 RAIL FREIGHT TRANSPORTATION 58
4.2 THE MODEL 62
4.3 RAIL MODELING 65
4.4 EMPTY FLOW ESTIMATION 83
4.5 BIBLIOGRAPHY 92

V **MODELING OF OTHER TRANSPORTATION MODES** 95
5.1 THE ROAD MODE 96
5.2 MODELING PORTS 98
5.3 NAVIGATION MODES 114
5.4 GENERAL CONSIDERATIONS 120
5.5 BIBLIOGRAPHY 122

VI **A NOTE ON MAXIMAL MULTICOMMODITY NETWORK FLOWS** 123
6.1 INTRODUCTION 123
6.2 THE PROBLEM 124
6.3 SOLUTION BY THE AUGMENTED LAGRANGEAN METHOD 125
6.4 SOME COMMENTS 127
6.5 COMPUTATIONAL EXPERIENCE 129
6.6 CONCLUSION 131
6.7 BIBLIOGRAPHY 132

VII **CALIBRATION OF DELAY FUNCTIONS** 133

7.1 NOTATION AND BASIC FORMULAE 135

7.2 DATA DESCRIPTION 139

7.3 CALIBRATION PROCEDURE 141

7.4 TRAIN AND DELAY APPROXIMATION 156

7.5 CONCLUSION 165

7.6 BIBLIOGRAPHY 166

7.7 APPENDIX - PROCEDURES 167

VIII **ON THE CONSIDERATION OF CAPACITY CONSTRAINTS IN THE STAN MULTI-MODE MULTI-PRODUCT ASSIGNMENT PROCEDURE** 173

8.1 THE NUMERICAL PROBLEM CONSIDERED 174

8.2 POSSIBLE APPROXIMATIONS OF THE "CAPACITY" FUNCTION 179

8.3 OTHER SOLUTION APPROACHES 188

8.4 COMPARISONS 198

8.5 CONCLUSIONS 201

8.6 BIBLIOGRAPHY 202

IX **A NUMERICAL APPROXIMATION PROCEDURE TO COMPUTE PARTIAL DERIVATIVES OF COST FUNCTIONS** 203
9.1 GRAM POLYNOMIALS 203
9.2 EQUIDISTANT OBSERVATIONS 204
9.3 CALIBRATION OF m AND n 205
9.4 THE APPROXIMATION FORMULA FOR THE DERIVATIVE 230
9.5 CHOICE OF THE STEP h 230
9.6 SOME FINAL TESTS 232
9.7 PERFORMANCE FOR STAN FUNCTIONS 232
9.8 CONCLUSIONS 233
9.9 BIBLIOGRAPHY 247

CHAPTER ONE

PROJECT SCOPE AND DELIMITATIONS

1.1 PROJECT SCOPE

The principal aim of this collaborative project between PUC-RJ, CRT(Université de Montréal) and GEIPOT was to develop the means and methods for evaluating contemplated changes in the transportation infrastructure that is used for freight movements in Brazil. The planning methods envisaged and the data base that support these methods were to be implemented in Brazil and provide the capability for carring out strategic planing exercises on a continuing basis.

1.2 STRATEGIC PLANNING

The strategic level of planning implies a medium to long term time horizon and a rather aggregate level of detail for the representation of the transport services provided. The contemplated alternatives represent usually anticipated (or predicted) changes in the origin to destination demands for the commodities considered and major changes to the transportation infrastructure. At the strategic level of planning, the aim is to obtain an overall evaluation of the movements of freight by most (or all) commodities on most (or all) of the modes available; thus tactical decisions (such as when to schedule the trains that are operated) are not considered at this level.

The planning periods for strategic planning alternatives are usually between three months and one year. The capacities of the transportation infrastructure are usualy given for each link and transshipment facility in terms of the number of vehicles of the particular mode considered. The anticipated demands for freight shipments are usually specified in tons and converted into vehicle (wagon) equivalent. The costs considered are usualy the time delay on links and through terminals, the operating cost of the supplier of transportation services and the fares paid by the shippers. The time delays, operating costs and fares reflect average values over the planning period.

1.3 METHODOLOGY FOR PREDICTING FLOWS, SERVICES LEVELS AND COSTS

The methods used to simulate flows, service levels and costs for a given alternative take as input the origin to destination demands, the allocation of products to particular modes (mode choice), the transportation infrastructure with link capacities, transshipment terminals with their delay and cost characteristics, and predict the way in which the demands give rise to the link flows for each mode and to transshipment flows through terminals. The service levels from origins to destinations are derived from link and terminal delays and the costs from origins to destinations are similarly evaluated by the appropriate methodology for each mode: they are computed either as the sum of link costs or estimated exogenously for each origin to destination shipment.

When the services of a given mode are limited (such as the train), the method of simulating the flows considers explicity the link capacities. The resulting models are multi-product multi-commodity network flow problems. When the services of a given mode are not limited, for all practical purposes, the method of simulating the flows allocates the origin to destination demands to routes of least delay and cost.

The origin to destination demands used for a contemplated alternative are estimated for the planning periods considered by using projections of known demand matrices, based on future productions and attractions by either manual methods or matrix balancing methods (which are adequatly chosen trip distribution methods).

1.4 THE DATA BASE

The data base required for supporting the strategic planning method outlined above requires the following principal elements, for each alternative considered:

- The transportation network for each mode, consisting of nodes, links, transshipment points;
- The capacities, time delay and cost characteristics of the network elements for each mode;
- The origin to destination demands for products considered;
- The tariffs for shipping a given product on a given mode, from an origin to a destination;
- The energy consuption characteristics of the vehicles of each of the considered modes.

A new alternative is defined by changes in any one of the data base elements outlined above, to reflect an anticipated future situation.

1.5 ALTERNATIVE COMPARAISON AND EVALUATION

Usualy, an existing situation in described by one of the scenarios that is contained in the data base. Future situations give rise to different alternatives; the flows, service levels and costs are simulated for each scenario which correspond to an alternative and then compared to the scenario which corresponds to the existing situation. A variety of local and global evaluation criteria may be computed from the differences of flows, service levels and costs and cost/benefit analyses may be carried out by considering the investment for a future alternative.

CHAPTER TWO

SYSTEM MODELING

This chapter is dedicated to the presentation of the general modeling concepts used in STAN to represent national or regional multi-mode multi-commodity transportation systems. We also take this opportunity to justify some of the choices that were made, and indicate some of the challenges inherent to modeling freight flows in a strategic planning environment. Detailed descriptions of the representation of specific modes representations are given in Chapters 4 and 5. Here, we describe the general system modeling concepts, which also serve to guide the accomplishement of future applications.

The physical network infrastructure, its characteristics, costs and operating policies, are represented by a network model that allows the transportation of several products by several modes.

A *product* is any commodity (collection of similar products), goods or passengers, for which there is a demand for its transport and thus generates flows on the links that are available for its movement. Each product is defined by a number of characteristics that describe its physical attributes and the way that it is being transported, such as the specification of the particular *vehicle* used to transport the product by *each one* of the modes that may move it.

Sixteen different products may be concurently defined in a STAN data bank. For reasons of computational efficiency, only six may be simultaneously used in a given assignment. Therefore, other than the usual aggregations of individual commodities to obtain products classes, one may need to further aggregate products in order to be able to *simultaneously* assign all the products on the network. While performing an aggregation of products, one should try to lump together products with similar transportation characteristics. There are many ways in which such an aggregation can be carried out. For instance, if most products are aggregated into one supercommodity, in order to analyse in detail the assignment of a few, chosen products, the aggregation operation has then a particular aim.

A *mode* is a means of transportation that has its own characteristics, such as vehicle type and capacity for each product, as well as specific cost functions. Modes may be differentiated by the type of infrastructure they use: rail, road, navigation, air, etc. The concept of *mode* is much richer, however, and may be used to model a broader spectrum of possibilities.

Modes may be used to model various "traction" modes (e.g. electric and diesel for rail or coastal and ocean navigation) and types (e.g. normal, narrow and large rail track gauges) used on the same infrastructure, or any combination thereof. Ownership, managerial or geographical differences may be identified by an appropriate specification of the modes. Special facility types, such as ports, may also be adequately represented by a particular mode. Specific examples may be found in Chapter 5.

Yet. despite the rich possibilities offered by this modeling approach, it is only too easy to burden the model with details that, although important in actual operations, should not be part of a strategic planning model. For instance, one may represent transportation services, such as train services, as different modes. This would yield an unmanageble number of modes for most realistic applications, without adding significant information to the results, such as the total utilization of a particular facility by the traffic flow, which are important for strategic decision making.

For each mode, a *typical* or *average* vehicle is defined in terms of the its dead weight and average transported tonnage, as well as, when applicable, by the weight (or length, or composition) of the corresponding convoy (train, multi-trailer truck, multi-barge convoy, etc.). These characteristics are obtained by considering the corresponding features and utilization of the real vehicles employed to transport the product on the particular mode. The "weighted average" or the "most frequently used" vehicle are the most appropriate for specifying this parameter. Examples of computations used to derive these parameters for the various modes and products are given in Chapters 2, 3 and 4 of Part IV.

The *base network* is the network that consists of the nodes, links and modes that represent all the physical movements possible on the available infrastructure. The network model that we have chosen defines a link as a triplet (origin node, destination node, mode allowed on the arc). Parallel links are used to represent the situation where more than one mode is available for transporting goods between two adjacent nodes.

In order to motivate our network modeling choice, consider the following simple example, which is shown in Figure 1. It consists of three modes: road transport, diesel rail and electrified rail. Cities A and B are served by all modes; A and C are served by diesel rail and road, whereas B and C are served only by road. In order to simplify the presentation, no intermodal transfers are permitted.

The most compact representation of this network is to connect all cities by directed links and to allocate the modes as link attributes (see Figure 2). Thus, link (A,B) has all the modes permitted, link (A,C) has the diesel and road modes, while link (B,C) has only the road mode.

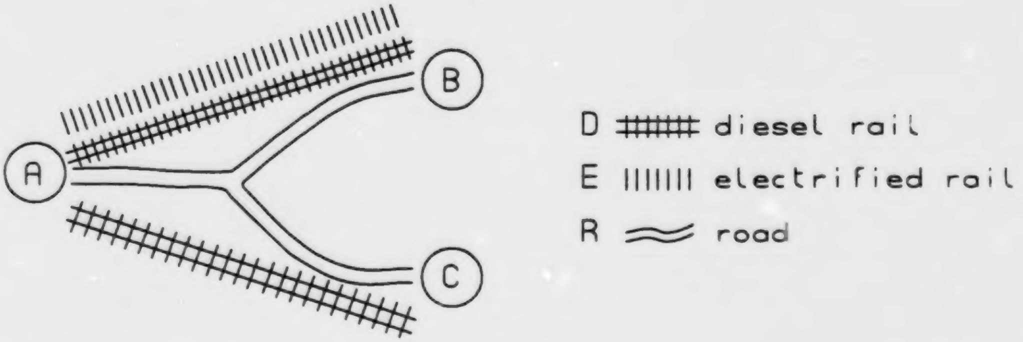


Figure 1 Physical Network

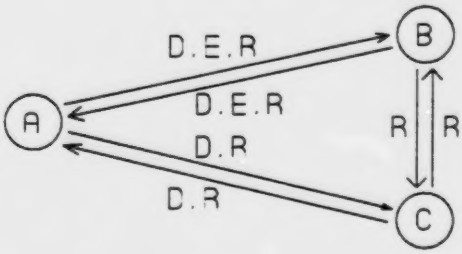


Figure 2 Compact Representation

This network representation has some major disadvantages for the model that we seek. If a single flow is associated with a link, it must be the total flow on the link and the flows specific to each mode are not kept explicitly. If multiple flows, one for each mode, are associated with each link, then these may vary in number from link to link, depending on the number of modes associated with a given link. In addition, the physical difference of the infrastructure modeled is not explicit in such a representation and the specification of cost and delay functions for each mode poses the same problem as that mentioned above for the flows for each mode.

In order to overcome these drawbacks, it is necessary to choose a network representation which permits easily the identification of flow of goods by mode, as well as cost and delay functions by mode. This is equivalent to keeping explicit "copies" of the network, one for each mode defined in a particular application. This type of network representation becomes prohibitive if large networks with many modes need to be considered.

An elegant way of achieving the aim of efficient representation of the multimodal network is to permit parallel links between each node, one for each mode permitted. In this way, the network model resembles the physical network, since the rail and road infrastructures are physically different. Also, if on a physical link, such as the rail facility, there are two different types of services, such as the diesel and electric train traction types of the exemple described in Figure 1, a separate link may be attributed to each service, since they would have different cost and delay functions. For this example, the network with parallel links is described in Figure 3.

In this representation, the mode is an integral part of the network and is not merely an attribute of the link. The parallel links introduced in the network may lead to difficulties in implementing network algorithms, unless parallel links between adjacent nodes that represent different modes are structurally different. This is accomplished by defining a link as a triplet: origin, destination, mode.

Nodes generally represent physical locations (cities, ports, junction points, etc.), and are used to insure that the base network realisticaly mirrors the layout of the infrastructure. Some nodes play special roles in the network model: those where intermodal transshipments may take place and centroids.

Once the network representation chosen, it is necessary, in order to model intermodal shipments, to permit mode transfers at certain nodes of the network, and to

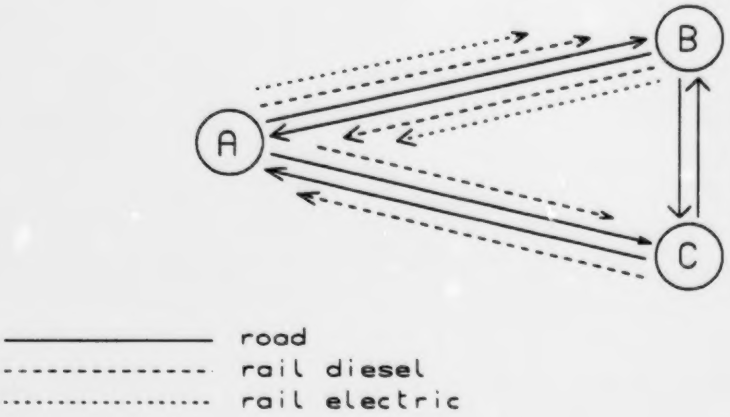


Figure 3 Network Representation Chosen

associate to these transfers the appropriate cost and delay functions. This may be accomplished by expanding (or exploding) a node where transfers occur by adding as many nodes as there are arcs entering and exiting the node and by adding transfer links between these nodes to form a bipartite graph. An example is given in Figure 4.

The explicit explosion of nodes would increase the number of links and arcs of the network. Also, if carried out explicitly, the new nodes do not represent the physical network anymore and the transfer arcs, representing mode to mode movements, are no longer unimodal as the other arcs of the network.

The representation chosen for transfers does not require the explicit modification of the network. Transfers are represented implicitly by a pair of arcs, one reaching the node and the other leaving the node. Transfer arcs are not added to the base network; rather the transfers permitted at each node may be displayed or listed by referring only to the pairs of arcs defining the transfers at a given node.

For strategic planning purposes, the demand for transportation cannot be individually considered for each possible node of the network. This would generate

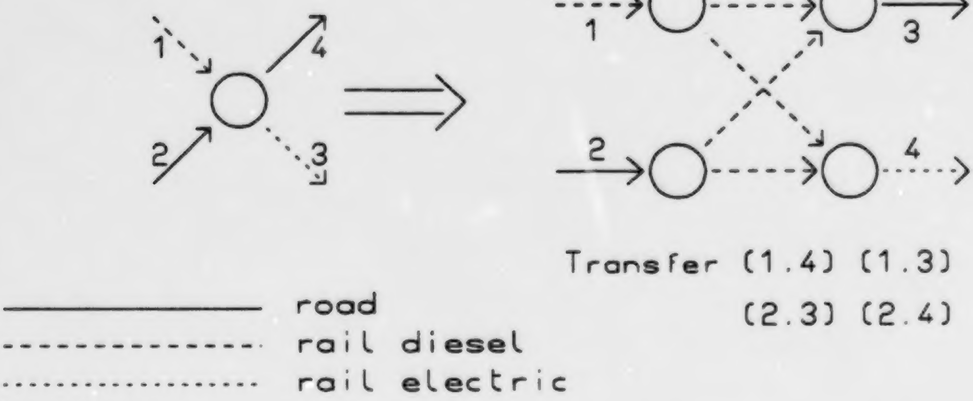


Figure 4 Explosion of a Node

unmanageably large origin/destination matrices and would explode both the model size and the computational time required by the algorithm, without improving the forecasts made. *Zones*, that contain several nodes in a contiguous geographical area, are defined and the transportation demand is aggregated accordingly. Zones may be more or less large according to the particular application. Each zone is then represented in the network by a special node, called a *centroid*, where all traffic originates and terminates, for all the economic activities at all the nodes included in the zone. The origin/destination transportation demand matrices are defined according to these centroids.

Centroids are connected to the rest of the network by links called *connectors*. Connectors have the same attributes and parameters as regular links and, by the adroit manipulation of cost functions, they may be used to model the specific entry or exit points of certain products into or out of certain zones. This is further discussed in Chapter 4.

The flows of products on the multi-modal network are the induced flows by the demande for these products on the links and transfers of the network. These flows are the decision variables of the network optimization model. For compatibility reasons, all flows are measured in the same weight units, which are usually tons. However, in order to adequately model the costs and the congestion phenomena associated with frieght movements, the corresponding number of vehicles and, if needed, of convoys, is locally evaluated on each link for each product flow. The estimation is performed, on each link and transfer of the paths used to move the product from origins to destinations on the network, by using the above mentioned characteristics of the typical vehicles associated to the product and mode. We achieve in this way the double objective of insuring the compatibility among modes, by a uniform modeling of the flow on the network, and a realistic representation of the utilization of the infrastructure and evaluation of the transportation cost.

Congestion phenomena arise naturally in a transportation system. There are modes, such as pipelines, ships and port grain loading/unloading equipment, where strict capacities limit the flow of goods through various facilities. There are also modes where an increasing number of vehicles (or convoys) compete for the use of the same infrastructure, and thus increase the passing-through time to unacceptable levels well before the transportation capacity is reached. Rail lines or roads passing through or near large urban centers are typical examples of this situation. Thus, congestion is due to a limited infrastructure capacity and too many vehicles attempting to simultaneously use the infrastructure due to a high demand. Congestion has a direct and major impact both on the levels of service and the cost of the transportation activities and thus has to be considered in any realistic model.

The approach that we have adopted does not explicitly consider the link or transfer capacities, but rather integrates them into volume/delay functions that affect the time and cost of transportation. This enhances the efficiency of the assignment procedure and results in a better representation of the increase of the transportation delays due to higher traffic volumes. Such volume/delay functions may be calibrated as nonlinear delay functions on particular facilities, or as nonlinear approximations of linear functions subject to a capacity constraint. Rail lines and yards offer perfect examples of the first case and are discussed in detail in Chapters 4 and 7. Port facilities and roads near large urban centers fall into the second class and appropriate function calibration methods are presented in Chapters 5 and 8.

The network optimization model that is used to simulate the network flows in STAN is a nonlinear multi-mode multi-commodity assignment that minimizes the *total generalized system cost* while satisfying the usual flow conservation constraints. The justification of the choice of a normative model has been presented in Chapter 1, while detailed mathematical descriptions and analyses of the network model and of the assignment algorithm are given in Chapter 3.

The generalized system cost is computed, for each link and transfer of the network, as the weighted sum of an operating cost function, a delay (passing through time) function and an energy consumption function. For each product, a function, of any type, has the form of an *average cost function* that multiplies the total product flow passing through the given link or transfer. The average cost function measures the cost of moving one unit of product flow through the link or transfer and, usually, depends on the physical characteristics of the infrastructure and the total commodity traffic.

The nonnegative weights, appearing in the total generalized system cost, reflect the relative importance of each cost component for a particular application, while the delay and energy consumption functions ensure the translation of time and energy factors into monetary units. This requires a few clarifications.

It is normally understood that operating costs include not only the costs directly associated with handling products and vehicles, but also "expenses" related to time, such as depreciation costs or crew costs, and to the consumption of energy. Consequently, one may perform an assignment in STAN with all costs accounted for in the operating cost function, the remaining two components receiving then zero weights.

Yet, it may be just as appropriate, for a wide variety of strategic scenarios, to perform assignments where only the time or the energy components are taken into account. Alternatively, the relative weights of the cost components may be calibrated to reflect different national priorities, such as energy conservation, instead of simply modeling an accounting function. One may also represent trade-offs among various cost components that more strongly affect various elements of the transportation system. Thus, for example, different transportation priorities among products could be translated into different unit *delay costs*, and the corresponding product delay costs could then be compared to the operating cost of the transportation system. These considerations amply demonstrate that the generalized cost function approach is much richer, offering a wider range of modeling possibilities, than a more classical method, where a single "monolithic" objective function is used to account for all the cost elements.

The use of STAN does not impose any definite rule for the choice of the cost elements to consider ; traction, maintenance, crews, fuel and lubrication, etc., may all be relevant in some applications and should then be accounted for. One should, however, pay particular attention that the **same** cost elements be considered for all modes and, additionnally, that each cost element be measured in the **same** units for all modes. This general rule, if applied consistently, also settles the question of the possible inclusion of the "fixed" costs into the computation: if for some modes (e.g. rail) these costs are included, whereas for others (e.g. motor carriers) they are not, the rule implies that they should be discarded in the computations for the multi-modal formulation. The uniform treatment of modes is more relevant to the accuracy of the results, than a conformistic translation of accounting procedures.

The general modeling concepts exposed in this chapter have been applied throughout the network model imbedded in STAN, as well as in the applications presented in this report. They represent sound modeling choices, insuring adequate trade-offs between the realism of the model, the efficiency of the solution algorithm and appropriate respose to policy sensitive planning issues for national or regional strategic transportation planning.

CHAPTER THREE

THE MULTI-MODE MULTI-PRODUCT NETWORK ASSIGNMENT MODEL AND ALGORITHM

The prediction of multicommodity freight flows over a multi-modal network has attracted much interest in recent years. In contrast to urban transporation, where the prediction of passenger flows over multi-modal networks has been studied extensively and many of the research results have been transferred to practice (see, for instance, Florian, 1984), the study of freight flows at the national or regional level, perhaps due to the inherent difficulties and complexities of such problems, received less attention.

The class of models that was well studied in the past for prediction of interregional flows is the *spatial equilibrium model* and its variants. This model, stated initially by Samuelson (1952) and extended by Takayama and Judge (1964,1970) then by Florian and Los (1982), Friesz, Tobin, and Harker (1983), has been used extensively for analyzing interregional commodity flows. This class of models determines simultaneously the flows between *producing* and *consuming* regions as well as the *selling* and *buying* prices. The transportation network is usually modeled in a simplistic way (bipartite networks) and these models rely to a large extent on the *supply* and *demand* functions of the producers and consumers respectively. The calibration of these functions is essential to the application of these models and the transportation costs are unit costs or may be functions of the flow on the network. There have been so far few multicommodity applications of this class of models, with the majority of applications having been carried out in the agricultural and energy sectors in an international or interregional setting. It is not this class of models which is the main topic of this chapter.

The class of models which we consider are network models which enable the prediction of multicommodity flows over a multi-mode network, where the physical network is modeled at a level of detail appropriate for a nation or a large region, and represents the physical facilities with relatively little abstraction. The demand for the transportation services is exogenous and may originate from an input-output model, if one is available, or from other sources, such as observed demand or scaling of observed past demand. The choice of mode, or subsets of modes used, is exogenous and intermodal shipments are permitted. In this sense, these models may be integrated with econometric demand models as well. The emphasis is on the network representation and the proper modeling of congestion effects in a static model that serves for comparative static analyses or for discrete time multiperiod analyses.

The Harvard Model (Kresge and Roberts, 1971), which is probably the first published predictive freight network model of the type that interests us, resorted to a fairly simple "direct link" representation of the physical network and congestion effects were not considered. Later, the Multi-State Transportation Corridor Model (McGinnis et al., 1981; Jones and Sharp, 1977; and Sharp, 1979) went a step further in representing an explicit multi-modal network, but without any consideration of congestion. The Transportation Network Model (Bronzini, 1980) does not consider congestion effects either. The first model that considers congestion phenomena in this field, is the Freight Network Equilibrium Model (Friesz, Gottfried and Morlok, 1986). This is a sequential model which uses two network representations: an aggregate network that is perceived by the users, which serves to determine the *carriers* chosen by the *shippers*, and then more detailed separate networks for each carrier, where commodities are transported at least total cost.

In an effort to combine the variable demand modeling approach of spatial equilibrium models with a detailed description of the behavior of shippers and carriers, in recent articles, Harker and Friesz (1986a, 1986b) present a theoretical approach and mathematical formulations that are yet to be tested in a particular application.

The class of models that we present in this chapter do not consider shippers and carriers as distinct actors in the decisions made in shipping freight. The simplifying assumption is made that the goods are shipped at least cost, even though the shipment of freight is governed by a variety of micro-circumstances that prevent the achievement of total cost minimization. The justification for this assumption is that, at a strategic level, predicting freight flows based on this assumption, when reliable estimates of modal demand are available, is satisfactory for the purpose of scenario comparisons, when the investments considered are of large magnitude.

The *multi-modal* aspects of a national transportation system are accounted for in the network representation chosen. A link of the multi-modal network is defined by its origin and destination nodes and a single mode. Parallel links are allowed between two adjacent nodes, one for each mode available to transport goods between them. The intermodal transfers at a node of network are modeled as link to link permitted movements. Appropriate cost functions may be associated with links and intermodal transfers.

The *multi-product* aspects of a national transportation system are accounted for in the formulation of the predictive model and is taken advantage of in the solution procedure. The algorithm developed for this problem exploits the natural decomposition by product and results in a Gauss-Seidel like procedure.

The multi-mode multi-product model is formulated in the most general way, permitting in principle nonconvex and assymmetric cost functions. Nevertheless, certain assumptions that are made on the structure of the cost functions simplify the problem and permit the solution of large size problems in reasonable computational times.

This chapter, which is mainly based on Guélat's (1986) work, is organized in the following way. First, the network representation chosen to integrate modes and transfers is described in detail; then the multi-mode multi-product assignment model is formulated and the solution algorithm adopted is presented; next, two salient aspects of the model are analyzed in detail, namely the form of the average and marginal cost functions and the computation of shortest paths with intermodal transfer costs on the network representation chosen; finally, some computational results obtained with this algorithm are presented.

3.1 THE NETWORK REPRESENTATION

The physical network infrastructure represented by the network model chosen supports the transportation of several products on several modes. A *product* is any commodity (collection of similar products), goods or passengers, that generates a link flow specifically associated with it. A *mode* is a means of transportation that has its own characteristics, such as vehicle type and capacity, as well as a specific cost function.

The *base network* is the network that consists of the nodes, links and modes that represent all the physical movements possible on the available infrastructure. The model that we have chosen defines a link as a triplet (i, j, m), where i is the origin node, $i \in N$, N is the set of nodes of the network, j is the destination node, $j \in N$, and m is the mode allowed on the arc, $m \in M$, where M is the set of modes available on the network. Parallel links are used to represent the situation where more than one mode is available for transporting goods between two adjacent nodes. In order to motivate our network modeling choice, we present the following simple example, which is shown in Figure 1. It consists of three modes: road transport, diesel rail and electrified rail.

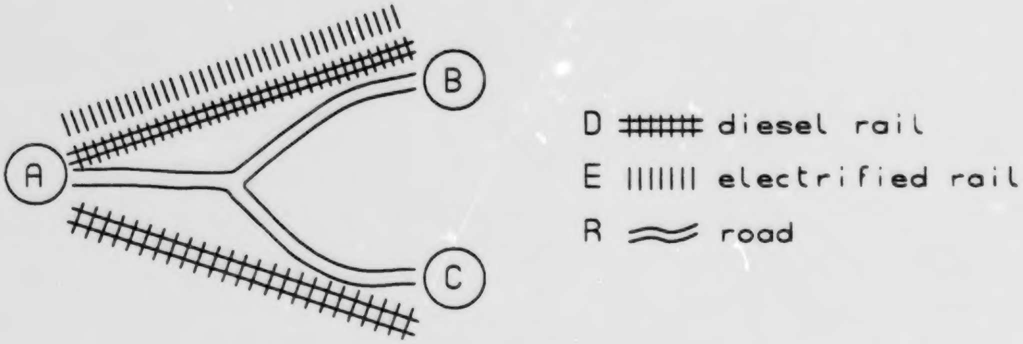


Figure 1 Physical Network

Cities A and B are served by all modes; A and C are served by diesel rail and road, whereas B and C are served only by road. In order to simplify presentation, no intermodal transfers are permitted.

The most compact representation of this network is to connect all cities by directed links and to allocate the modes as link attributes. Thus, link (A,B) has all the modes permitted, link (A,C) has the diesel and road modes, while link (B,C) has only the road mode. This network representation illustrated in Figure 2a, has some major disadvantages for the model that we seek. If a single flow is associated with a link, it must be the total flow on the link and the flows specific to each mode are not kept explicitly. If multiple flows, one for each mode, are associated with each link, then these may vary in number from link to link, depending on the number of modes associated with a given link. In addition, the physical difference of the infrastructure modeled is not explicit in such a representation. The specification of cost and delay functions for each mode poses the same problem as that mentioned above for the flows for each mode.

In order to overcome these drawbacks, it is necessary to choose a network representation which permits easily the identification of the flow of goods by mode, as well as cost and delay functions by mode. This is equivalent to keeping explicitly "copies" of the network, one for each mode defined in a particular application. This type of network representation becomes prohibitive if large networks with many modes need to be considered.

An elegant way of circumventing this problem is to permit parallel links between each node, one for each mode permitted. In this way, the network model resembles the physical network, since the rail and road infrastructures are physically different. Also, if on a physical link, such as the rail facility, there are two different types of services, such as the diesel and electric train services of the example described in Figure 1, a separate link may be assigned to each service, since they would have different cost and delay functions. For this example, the network with parallel links is described in Figure 2b.

In this representation, the mode is an integral part of the network and is not merely an attribute of the link. The parallel links introduced in the network, as may be seen in Figure 2b, may lead to difficulties in implementing network algorithms, unless parallel link between adjacent nodes that represent different modes are structurally different. This is accomplished by defining a link as a triplet (origin, destination, mode).

Once the network representation chosen, it is necessary, in order to model intermodal shipments, to permit, and associate the appropriate costs and delays, for mode transfers at certain nodes of the network. This may be accomplished by expanding (or exploding) a node where transfers occur, by adding as many nodes as there are arcs entering and exiting the node and by adding transfer links between these nodes to form a bipartite graph. An example is given in Figure 3.

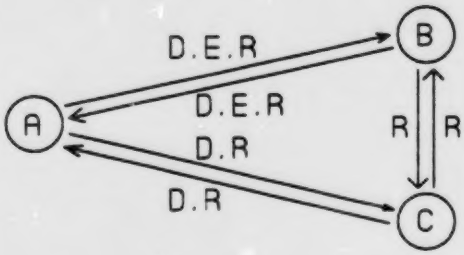


Figure 2a Compact Representation

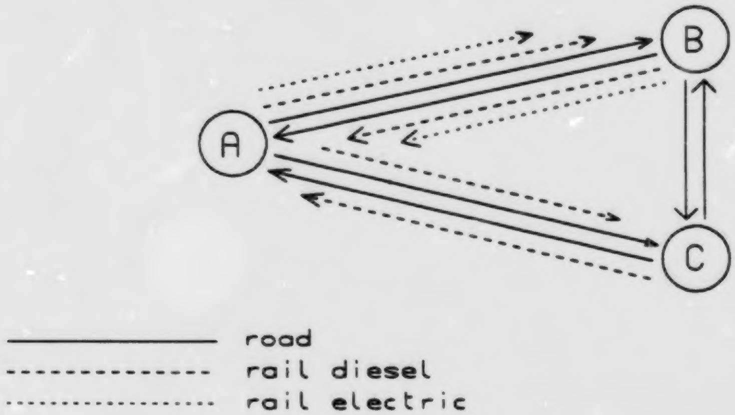


Figure 2b Network Representation Chosen

The explicit explosion of nodes would increase the number of links and arcs of the network; also, if carried out explicitly, the new nodes do not represent the physical network and the transfer arcs represent a mode to mode transfer, so they are not unimodal as the other arcs of the network.

The representation chosen for transfers does not require the explicit modification of the network. Transfers are represented implicitly by a pair of arcs, one reaching

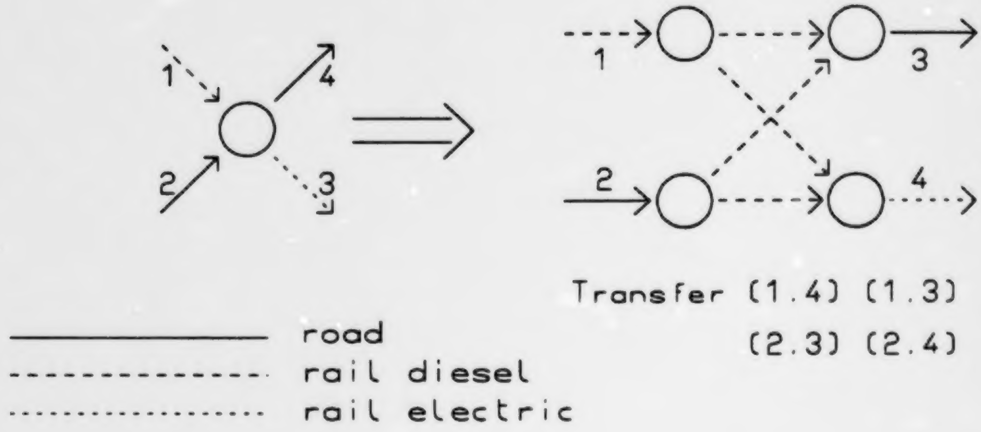


Figure 3 Explosion of a node

the node and the other leaving the node. Transfer arcs are not added to the base network; rather the transfers permitted at each node may be displayed or listed by referring only to the pairs of arcs defining the transfers at a given node.

3.2 THE MULTI-MODE MULTI-PRODUCT MODEL

In the context of strategic planning of freight flows on a national or regional scale, the most efficient use of the transportation infrastructure is to transport the freight at least total (generalized) cost. Even though it is reasonable to assume that, even in countries where a central authority controls and regulates the shipment of goods, a variety of micro-circumstances in fact prevent the precise achievement of the goal of shipping at least cost, the model that we formulate is based on the objective of minimizing total costs. The notion of cost is central to our model and we interpret cost in the most general way, in the sense that it may have different components, such as monetary cost, delay, energy consumption, etc.

The network that we consider consists of a set of nodes N, a set of arcs A, $A \subseteq N \times N$, a set of modes M and a set of transfers T, $T \subseteq A \times A$. We denote their cardinality, n_N, n_A, n_M and n_T respectively. With each arc a, $a \in A$, we associate a *cost function* $s_a(.)$ which depends on the volume of goods on the arc, or possibly, on the volume of goods on the other arcs of the network. Similarly, a cost function $s_t(.)$ is associated with each transfer $t \in T$.

The products transported over the multi-modal network are denoted by index p, $p \in P$, where P is the set of all products considered, which is of cardinality n_P. Each product is shipped from origins o, $o \in O \subseteq N$, to destinations d, $d \in D \subseteq N$, of the network. The demand for each product and for all origin/destination (O/D) pairs is specified by a set of O/D matrices for the corresponding subsets of modes. It is assumed that the mode choice is determined exogenously. Let $g^{m(p)}$ be the demand matrix associated with product $p \in P$, where $m(p)$ is a subset of modes that belong to $M(p)$, the set of all subsets of modes that are used to transport product p.

The flows of product p on the multimodal network is denoted by v^p and consists of the induced flows of this products on links and transfers :

$$v^p = \begin{pmatrix} (v_a^p), & a \in A \\ (v_t^p), & t \in T \end{pmatrix}$$

The flow of all the products on the multimodal network is denoted by $v = (v^p)$, $p \in P$, and is a vector of dimension $n_P(n_A + n_T)$.

The average cost functions $s_a^p(v)$, on links, and $s_t^p(v)$, on transfers, correspond to a given flow vector v. The average cost functions for product p are denoted, similar to the notation used for the flow v^p, s^p, $p \in P$, where

$$s^p = \begin{pmatrix} (s_a^p), & a \in A \\ (s_t^p), & t \in T \end{pmatrix}$$

and $s = (s^p)$, $p \in P$, is the vector of average cost functions of dimension $n_P(n_A \times n_T)$.

The total cost of the flow on arc a, $a \in A$, for the product p, $p \in P$, is the product $s_a^p(v)v_a^p$; the total cost of the flow on transfer t, $t \in T$, is $s_t^p(v)v_t^p$. The total cost of the flows of all products over the multi-modal network is the function F that we seek to minimize

$$F = \sum_{p \in P}(\sum_{a \in A} s_a^p(v)v_a^p + \sum_{t \in T} s_t^p(v)v_t^p) \qquad (= s(v)^T v) \tag{1}$$

over the set of flows which satisfy the conservation of the flow and nonnegativity constraints. In order to write these constraints for the multi-product multi-mode network defined above, the following notation is used. Let $K_{od}^{m(p)}$ denote the set of paths that lead from origin o, $o \in O$, to destination d, $d \in D$, by using only modes of $m(p) \in M(p)$, $p \in P$. The conservation of flow equations are then

$$\sum_{k \in K_{od}^{m(p)}} h_k = g_{od}^{m(p)}, \qquad o \in O, d \in D, m(p) \in M(p), p \in P \tag{2}$$

where h_k is the flow on path k. The nonnegativity constraints are

$$h_k \geq 0, \qquad k \in K_{od}^{m(p)}, o \in O, d \in D, m(p) \in M(p), p \in P \tag{3}$$

Let Ω be the set of flows v that satisfy (2) and (3). Since the conservation of flow equations are stated in the space of path flows, for notational convenience, the specification of Ω requires the relation between arc flows and path flows, which is

$$v_a^p = \sum_{k \in K^p} \delta_{ak} h_k, \qquad a \in A, p \in P \tag{4}$$

where

$$K^p = \bigcup_{m(p)\in M(p)} \bigcup_{o\in O} \bigcup_{d\in D} K_{od}^{m(p)}$$

is the set of all paths that may be used by product p, and

$$\delta_{ak} = \begin{cases} 1 & \text{if } a \in k \\ 0 & \text{otherwise} \end{cases}$$

is the indicator function which identifies the arcs of a particular path. Similarly, the flows on tranfers are

$$v_t^p = \sum_{k\in K^p} \delta_{tk} h_k, \qquad t \in T, p \in P \tag{5}$$

where

$$\delta_{ak} = \begin{cases} 1 & \text{if } t \in k \\ 0 & \text{otherwise .} \end{cases}$$

The transfer t belongs to the path k if the two arcs that define the transfer belong to it.

In conclusion, the system optimal multi-product, multi-mode assignment model consists of minimizing (1) subject to (2)-(3) with the definitional constraints (4)-(5).

The model is sufficiently general to be adapted for different ways of specifying the demand. Even though typical applications of the model are likely to be carried out after an "a priori" mode choice calculation, which would allocate the demand for a product g^p to a set of mode subsets, it is equally possible to permit the demand for a product to be transported over all the allowed modes, that is $m(p)$ is the set of all modes of the network and $M(p)$ has a single component, which in this case is $m(p)$. Also, the model is flexible in the specification of intermodal movements. The mode to mode transfers may be restricted to occur only at specific nodes of the network, and only between specific modes.

We turn our attention next to develop a solution algorithm for this model.

3.3 A SOLUTION ALGORITHM

The problem that we consider may be stated in a compact form as

$$\text{Min } F(v), \quad \text{subject to} \quad v \in \Omega \qquad (P)$$

Since constraints (2)-(5) are linear, Ω is a polytope which, if the multi-modal network is strongly connected, is nonempty. Ω is compact as well, if the cost functions that we consider are strictly positive, hence the paths considered do not contain cycles.

It is assumed that $F(v)$ is a convex function, once differentiable on an open space that contains Ω. These assumptions are satisfied, for example, by average arc (transfer) cost functions of the type

$$s_a^p = \sum_i \alpha_i(\beta_i + v_a^p)^{z(i)}, \qquad \alpha_i > 0,\ \beta_i > 0,\ 0 \leq z(i) < \infty.$$

As demonstrated by this example, the average cost functions need not be convex $(0 < z(i) < 1)$ for $F(v)$ to be convex.

The size of the problems that are likely to occur in specific applications $(n_p(n_A + n_T) \sim 100,000-150,000)$ mitigates the use of second order algorithms that require the use of the Hessian of the objective function or its approximation in order to solve (P). One of the most used first order methods is the linear approximation method (Frank and Wolfe, 1956). This method, which is widely used for solving large scale network equilibrium problems (see Florian, 1986) has a sublinear rate of convergence in the vicinity of the optimal solution. However, it has proved to be robust in applications which do not require a very precise solution. Its straightforward adaptation for the model that we seek to solve poses a problem of dimension: the vector v is of dimension $n_p(n_A+n_T)$ and so is the dimension of the feasible direction of descent generated by the linear approximation method. This may render the straightforward application of the linear approximation method impracticable on the most widely available computers.

The structure of the model suggests a natural decomposition by product. Ω is the direct product $\prod_p \Omega^p$, where Ω^p is the set of feasible flows of product p on the subnetworks $m(p) \in M(p)$. The descent direction of the linear approximation method is found by minimizing the linear approximation of the function, which in this case would be F^p ($F = \sum_{p=1}^{n_p} F^p$); thus one would have to solve n_p subproblems in a cycle of the algorithm. The algorithm that we develop in this section is based on this observation, and may be characterized as a Gauss-Seidel - Linear Approximation Approach. For each product p, a descent direction is computed for v^p which is of dimension $(n_A + n_T)$. When considering a particular product, all the flow variables pertaining to the other products are kept at their previous value.

This approach resembles the block Gauss-Seidel method, except that the subproblem for each block is not solved exactly; rather, for each product p, a single iteration of the linear approximation algorithm is carried out. The statement of the algorithm is as follows:

Algorithm A

```
Step 0 - Initialisation
         Determine $v$      ( initial feasible solution)
Step 1 - Major cycle
         $v_{old} = v$,
         - For each $p \in P$;      carry out (1$p$).
         (1$p$) - Minor cycle
             Compute $c^p(v)$          (marginal costs)
             Find $w^p$                (extreme point found by shortest path
                                        computations)
             $d^p = w^p - v^p$         (descent direction)
             Compute $\lambda_p$       (optimal step size)
             $v^p = v^p + \lambda_p d^p$    (update flow for $p$).
Step 2 - Stopping criterion
         If ($v \neq v_{old}$), return to step 1.
```

Since the algorithm is stated in a rather compact form, some comments on the details of each step are in order. The minor cycle (1p) implements one iteration of the linear approximation method in the subspace of flows related to product p. A descent direction is found by minimising the linear approximation of F^p on Ω^p, which is

$$\text{Min} \quad \nabla F^p(v)y, \text{ subject to } y \in \Omega^p \qquad (\text{LP}_p).$$

Since $F(v) = s(v)^T v$ it follows that $\nabla F(v) = s(v)^T + v^T s'(v)$. Let the marginal cost be defined as $c(v) \doteq s(v) + s'(v)^T v$. The linear programming subproblem is

$$\text{Min} \quad c^p(v)^T y, \text{ subject to } y \in \Omega^p.$$

The solution of this problem, as is well known, is obtained by assigning the demand $g^{m(p)}$ on the shortest paths corresponding to the modes that belong to $m(p)$ computed with arc costs $c^p(v)$.

The optimal step size λ_p is obtained by a one-dimensional minimization of F in the direction $(0, ..., d^{p^T}, ..., 0)^T$ unless $c^p(v)^T d^p = 0$ in which case $\lambda_p = 0$.

An initial (extremal) solution may be obtained by carring out a major cycle with initial marginal costs corresponding to $v = 0$, that is $c(o)$ and by setting $\lambda_p = 1$ in each minor cycle (1p).

The algorithm is stopped when the flows v do not change after a major cycle was carried out. The current solution is optimal, as will be demonstrated in the following section. In pratice, the algorithm may be stopped after carrying out a predetermined number of major cycles or after a maximal execution time. In addition, the supposed convexity of the objective function F may be used to construct a stopping criterion based on the "relative gap" RG

$$\text{RG} = 100 * \frac{F(v) - BLB(v)}{F(v)}$$

where $BLB(v)$ represents the best lower bound obtained during the previous major cycles carried out. This lower bound is computed by solving the linear programming problem

$$\text{Min} \quad \nabla F(v)^T y \quad \text{subject to } y \in \Omega.$$

A lower bound on the optimal solution $F(v^*)$ is given by $F(v) + \nabla F(v)(y^* - v)$, where y^* is the optimal solution of the above problem. This lower bound may not be computed by using the shortest paths detained at each minor cycle (1p); it requires the solution of as many shortest paths problems as there are O/D matrices $g^{m(p)}$. As it rather "expensive" to compute, its computation would depend on the power of the computer available for solving the problem.

3.4 CONVERGENCE ANALYSIS OF THE ALGORITHM

In order to prove the global convergence of the algorithm presented in the previous section, we show that F is a descent function relative to algorithm A and that the algorithmic map, which we denote by A as well, is closed on Ω.

Lemma 1 : Let y be the solution obtained from v after carrying out a minor cycle $(1\bar{p})$. Then, $F(y) \leq F(v)$.

Proof :

$$y^p = \begin{cases} v^p + \lambda_p d^p & \text{if } p = \bar{p} \\ v^p & \text{otherwise.} \end{cases}$$

Then $y = v + \lambda_p D$, where $D = (D^p)_{p \in P}$ and

$$D^p = \begin{cases} d^p = w^p - v^p & \text{if } p = \bar{p} \\ 0 & \text{otherwise .} \end{cases}$$

Let $g(y) = F(v + \lambda D)$; by the assumptions made on F, g is continuously differentiable. Thus

$$\begin{aligned} g'(\lambda) &= \nabla F(v + \lambda D) D = c(v + \lambda D)^T D \\ &= \textstyle\sum_{p \in P} c^p(v + \lambda D)^T D^p = c^p(v + \lambda D)^T d^p \end{aligned}$$

and $g'(o) = c^p(v)^T d^p$.

Then :

- if $g'(0) = 0$, then $\lambda_p = 0$ in which case $y = v$ and $F(y) = F(v)$.

- if $g'(0) \neq 0$, then $g'(0) < 0$ by the definition of w^p; thus D is a feasible descent direction for F at v. By the definition of λ_p, it follows that $F(y) < F(v)$.

Now let $S \subseteq \Omega$ be the set of flows that satisfy the Kuhn Tucker conditions of the problem (P), or equivalently, the set of optimal solutions of P. If F is strictly convex, S is a singleton. The set S may be characterized by the variational form

$$S = \{v \in \Omega | \nabla F(v)(w - v) \geq 0, \forall w \in \Omega\}.$$

Let A : $\Omega \rightarrow \Omega$ be the (one to many) algorithmic map.

Lemma 2 : $v \in S$ if and only if $v \in A(v)$

Proof : First we prove that the condition is necessary.

$$\nabla F(v)(w - v) \geq 0 \quad , \forall w \in \Omega \quad \text{is equivalent to}$$
$$\textstyle\sum_{p \in P} c^p(v)^T(w^p - v^p) \geq 0 \quad , \forall w \in \Omega.$$

Consider $\bar{p} \in P$ and $y = (y^p)_{p \in P} \in \Omega$ such that $y^p = v^v$ if $p \neq \bar{p}$.
It follows that

$$c^{\bar{p}}(v)^T(y^{\bar{p}} - v^{\bar{p}}) \geq 0, \quad \forall y^{\bar{p}} \in \Omega^{\bar{p}}.$$

In particular, if $w^{\bar{p}}$ is the extremal flow generated in the linear subproblem of the minor cycle ($1\bar{p}$), it follows that $c^{\bar{p}}(v)^T(w^{\bar{p}} - v^{\bar{p}}) = 0$. Hence v will not be modified in any minor cycle and $v \in A(v)$.

The sufficiency of the condition is demonstrated by contraposition. Let $v \in \Omega - S$. There exists then a $w \in \Omega$ such that

$$\nabla F(v)(w - v) < 0 \quad \text{or} \quad \textstyle\sum_{p \in P} c^p(v)^T(w^p - v^p) < 0 .$$

It is impossible for $c^p(v)^T d^p = 0$ for all $p \in P$; but, in the proof of Lemma 1 it was shown that if $c^p(v)^T d^p < 0$ at a minor cycle the objective function F descreases strictly and as F cannot increase at a minor cycle, it follows that $F(y) < F(v)$ for all $y \in A(v)$. In particular v does not belong to $A(v)$.

Corollary 1 : F is a function of descent relative to A.

Proof : Let $y \in A(v)$. By applying Lemma 1 n_p times, one for each subproblem p, it follows that $F(y) \leq F(v)$. By Lemma 2, $F(y) < F(v)$ if $v \in \Omega - S$.

The second part of the convergence proof consists of the demonstration that the algorithm map A is closed on Ω. In order to do so, let $P = (p_1, \dots, p_{n_p})$ and A is decomposed accordingly:

$$\begin{aligned} A &= A^{p_{n_p}} \circ \dots\dots \circ A^{p_1} &&: \Omega \to \Omega \\ \text{where } A^p &= M \circ D^p &&: \Omega \to \Omega \end{aligned}$$

A^p is the algorithmic map that corresponds to a minor cycle, D^p represents the search for a descent direction and M the one-dimensional minimization. More formally, let

$$\begin{aligned} D^p \ :\ & \Omega \to \Omega \times U \\ & v \to D^p(v) \end{aligned}$$

with $D^p(v) = \{ \ (v,d) | d^T = (0, \dots, d^{p^T}, \dots, 0), d^p = w^p - v^p$, where

$w^p \in \text{ Arg Min}_{y \in \Omega^p} \nabla F^p(v) y \}$,

and $U = \Omega - \Omega = \{ y | y = y_1 - y_2 ; y_1, y_2 \in \Omega \}$.

Let also

$$\begin{aligned} M \ :\ & \Omega \times U \to \Omega \\ & (v,d) \ \to M(v,d) \end{aligned}$$

where

$$M(v,d) = \{ y | y = \begin{cases} v & \text{if } \nabla F(v) d = 0 \\ v + \lambda^* d & \text{otherwise, with } \lambda^* \in \text{ Arg Min}_{0 \leq \lambda \leq 1} F(v + \lambda d) \}. \end{cases}$$

Lemma 3 : D^p is a closed mapping on Ω.

Proof : Let $\{v\}_k$ be a sequence of Ω that converges to v. Let, on the other hand, $\{(v^{k,d^k})\}_k$ be a sequence of $\Omega \times U$ such that $(v^k, d^k) \in D^p(v^k)$ and which converges to (v, d). By the definition of $D^p(v^k)$, the direction d^k may be expressed as $y^k - v^k$ where $y^k = v^k$ except for the p'th component where $(y^k)^p = (w^k)^p$. Thus we have that

$$y^k = d^k + v^k \ , \ y^k \in \Omega \ .$$

The right term of the equation tends to $d + v \doteq y$; $y = v$ except for the p'th component. In order to show that (v, d) belongs to $D^p(v)$, it suffices to point out that y^p is the optimal solution of the linear subproblem

$$\text{Min } \nabla F^p(v)z, \text{ subject to } z \in \Omega^p.$$

$$\nabla F^p(v^k)(y^k)^p \leq \nabla F^p(v^k)z^p, \qquad \forall Z^p \in \Omega^p.$$

By invoking the continuity of ∇F it follows that

Then $\nabla F^p(v)y^p \leq \nabla F^p(v)z^p, \qquad \forall z^p \in \Omega^p.$

Lemma 4 : The map M is closed on $D^p(v)$, $\forall v \in \Omega$.

Proof : Let $(v, d) \in D^p(v)$ and $\{(v^k, d^k)\}_k$ be a sequence of $\Omega \times U$ which converges to (v, d). On the other hand, let $\{y^k\}_k$ be a sequence of Ω such that $y^k \in M(v^k, d^k)$, which converges to y. Consider first the case where $d \neq 0$. There exist then a K such that $d^k \neq 0$ for all $k \geq K$. In the following we consider that $k \geq K$.

How, $y^k = v^k + \lambda^{*k} d^k$,

hence $\lambda^{*k} = \frac{\|y^k - v^k\|}{\|d^k\|}$.

The term on the right hand side tends to

$$\frac{\|y - v\|}{\|d\|} \doteq \lambda^* \quad (\in [0, 1])$$

and hence $y = v + \lambda^* d$.

It remains to show that the λ^* above solves the one-dimensional minimization problem. Since $F(y^k) \leq F(v^k + \lambda d^k)$ for all $\lambda \in [0, 1]$, by using the continuity of F and by taking the limit for fixed λ, one obtains:

$$F(y) = F(v + \lambda^* d) \leq F(v + \lambda d), \ \lambda \in [0, 1] \ .$$

We consider now the case $d = 0$. Again

$$y^k - v^k = \lambda^{*k} d^k.$$

As $\lambda^{*k} \in [0, 1]$ for all k and as d^k tends to zero, the term on the right hand side tends to zero as well. By taking the limit one obtains that $y = v$.

Corollary 2 : A^p is closed on Ω.

Proof : $A^p : \Omega \rightarrow \Omega \times U \rightarrow \Omega$

$\Omega \times U$ is compact since Ω is a compact. By Lemmas 3 and 4 all the hypotheses of Lemma 4.2 of Zangwill (1969) are satisfied.

Corollary 3 : A is closed on Ω.

Proof : By applying corollary 2 to $A = A^{P_{n_p}} \circ \ldots\ldots \circ A^1$ and Lemma 4.2 of Zangwill (1969) the result follows.

The global convergence of the algorithm A follows from convergence Theorem A of Zangwill (1969) and Corollaries 1 and 3 above.

We turn our attention next to the average and marginal cost functions.

3.5 THE COST FUNCTIONS

In the minimization problem (P), defined in the section 3 of this paper, the average cost functions $s_a(v)$, $a \in A$, and $s_t(v)$, $t \in T$ may, in principle, depend on any or all components of the vector v. The Jacobian matrix, which appears in the definition of the marginal costs, may therefore be a full matrix, which usually implies lengthy evaluations of these costs. We consider next several restrictive and simplifying hypotheses on the structure of the average cost functions, which preserve nevertheless sufficient generality to model properly nonseparable average arc cost functions.

In its most general form, the marginal cost for transporting product $\bar{p}$ on arc $\bar{a}$ is given by the expression

$$c_{\bar{a}}^{\bar{p}} = s_{\bar{a}}^{\bar{p}} + \sum_{p \in P} \Big(\sum_{a \in A} \frac{\partial s_a^p}{\partial v_{\bar{a}}^{\bar{p}}} v_a^p + \sum_{t \in T} \frac{\partial s_t^p}{\partial v_{\bar{a}}^{\bar{p}}} v_t^p \Big) \tag{6}$$

and the marginal cost for transporting product $\bar{p}$ on transfer arc $\bar{t}$ is

$$c_{\bar{t}}^{\bar{p}} = s_{\bar{t}}^{\bar{p}} + \sum_{p \in P} \Big(\sum_{a \in A} \frac{\partial s_a^p}{\partial v_{\bar{t}}^{\bar{p}}} v_a^p + \sum_{t \in T} \frac{\partial s_t^p}{\partial v_{\bar{t}}^{\bar{p}}} v_t^p \Big) \tag{7}$$

The first simplifying assumptions that we make concern the dependence of the cost function on an arc on the flows on the transfers and vice versa.

Assumption 1 : $s_t^p(v)$, $p \in P$, $t \in T$, do not depend on the arc flows v_a^p , $p \in P$, $a \in A$.

With this assumption equation (6) simplifies to

$$c_{\bar{a}}^{\bar{p}} = s_{\bar{a}}^{\bar{p}} + \sum_{p \in P} \sum_{a \in A} \frac{\partial s_a^p}{\partial v_{\bar{a}}^{\bar{p}}} v_a^p \tag{8}$$

A similar simplification of equation (7) results from

Assumption 2 : $s_a^p(v)$, $p \in P$, $a \in A$ do not depend on the transfer flows v_t^p , $p \in P$, $t \in T$, which leads to the expression

$$c_{\bar{t}}^{\bar{p}} = s_{\bar{t}}^{\bar{p}} + \sum_{p \in P} \sum_{t \in T} \frac{\partial s_t^p}{\partial v_{\bar{t}}^{\bar{p}}} v_t^p . \tag{9}$$

Since transfers represent mode to mode movements it is reasonable to postulate

Assumption 3 : $s_t^p(v)$, $p \in P$, $t \in T$ do not depend on the flows on the other transfers $\bar{t} \in T$, $\bar{t} \neq t$.

With this assumption, (9) simplifies further to

$$c_{\bar{t}}^{\bar{p}} = s_{\bar{t}}^{\bar{p}} + \sum_{p \in P} \frac{\partial s_{\bar{t}}^{p}}{\partial v_{\bar{t}}^{\bar{p}}} v_{t}^{p} \tag{10}$$

Unfortunately, a similar assumption cannot be made for cost functions of the arcs. It is highly likely that two distinct modes (such as rail diesel, rail electric on a single track facility) share the same physical infrastructure between two nodes of the network. It is reasonable nevertheless to assume that the interactions are limited to links which are parallel, in both orientations, between two adjacent nodes. See Figure 4.

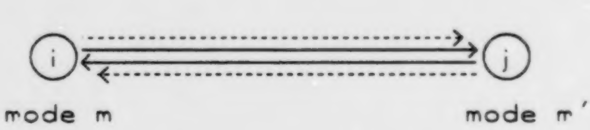


Figure 4 Parallel Links in Both Orientations

We introduce some additional notation. The arc $a = (i, j, m)$ connects node i to node j by mode m. Let M_a be the set of modes which is considered in the computation of the marginal cost of arc a. Let

$$A_a \doteq \{\tilde{a} \in A | \tilde{a} = (i, j, m') \text{ or } \tilde{a} = (j, i, m'), m' \in M_a\}$$

be the set of arcs that must be considered in the computation of the marginal cost of arc a; thus $s_a^p(v) = s_a^p(v_{\tilde{a}}^{\tilde{p}})$, $\tilde{a} \in A_a$, $\tilde{p} \in P$. If arc $\tilde{a} \in A_a$ then $\partial s_a^p / \partial v_{\tilde{a}}^{\tilde{p}} = 0$. This permits the simplification of the equation (8) to

$$c_{\bar{a}}^{\bar{p}} = s_{\bar{a}}^{\bar{p}} + \sum_{p \in P} \sum_{a \in H_{\bar{a}}} \frac{\partial s_a^p}{\partial v_{\bar{a}}^{\bar{p}}} v_a^p \tag{11}$$

where $H_{\bar{a}} = \{a \in A | \bar{a} \in A_a\}$. The set $H_{\bar{a}}$ can be obtained in general by scanning the arcs parallel to $\bar{a}$ in both orientations, which may be time consuming particularly if $H_{\bar{a}} = \{\bar{a}\}$ for the majority of the arcs. Therefore we make the additional hypothesis:

Assumption 4 : $H_a = A_a$ for all $a \in A$.

The marginal costs of the arcs may be writen then as

$$c_{\bar{a}}^{\bar{p}} = s_{\bar{a}}^{\bar{p}} + \sum_{p \in P} \sum_{a \in A_{\bar{a}}} \frac{\partial s_a^p}{\partial v_{\bar{a}}^{\bar{p}}} v_a^p \tag{12}$$

In the code which implements the algorithm of the preceeding section in STAN, the partial derivatives required in the computation of the marginal costs are carried out by a rather precise numerical approximation procedure (see Chapter 9). The analytic form of the cost functions may be rather complex, in particular for the rail mode (see Chapter 4) and, as a consequence, so are the analytic form of the partial derivatives and cost functions. Another consideration that motivates the numerical approximation procedure is that it obviates the specification of the partial derivatives by the user of the code, be they simple or complex.

The method that we chose to implement in order to compute the derivative of a function at a given value, is a polynomial approximation based on the least squares method, which appears to be superior to numerical interpolation (Kellison (1975), pp. 150-151). We have chosen to use the orthogonal polynomial of Gram, which was adapted for our purposes (see Chapter 9) starting from a code written by Casaletto and Rice (1978).

We turn our attention to the shortest path algorithm used to compute least marginal cost paths, which has a central role in the solution algorithm of the multimode multiproduct model.

3.6 THE SHORTEST PATH ALGORITHM

The multi-modal network that we use for the formulation of the multi-product assignment model contains transfer costs, which are similar to turn penalties in the context of urban transportation networks (see Potts and Oliver, 1972). Spiess (1984) proposed a link labeling procedure (which is implemented in the EMME/2 code) for computing shortest paths on urban road networks with turn penalties at intersections. Unlike urban road networks, in our model a path that contains the links of more than one mode can be constructed only if a mode to mode tranfer is permitted at a node of the base network that belongs to the path.

In order to state the algorithm that we use, we introduce the following notation. Let c_a, $a \in A$, and c_t, $t \in T$, be the "lengths" of the arcs and transfers. In algorithm A, these correspond to a set of current marginal costs. The shortest path algorithm finds the shortest path from an origin $o \in O$ to all destination $d \in D$ in the subnetworks defined by the subset of modes of $m(p) \in M(p)$. The shortest path is retraced by the means of pointers to the preceding link: b_d is the access link to destination d, b_a is the predecessor of link a; the length of the shortest paths are given by variables u_d and u_a, where u_a is the lenght of the shortest path from the origin o to link a inclusive of c_a; $\bar{A}$ represents the set of links which were examined (labeled) but do not yet have permanent labels.

Shortest path algorithm with transfers

```
Step 0 -  Initialization
          Lengths          : $u_d = \infty, d \in D, u_o = 0; u_a = \infty, a \in A$
          Predecessors     : $b_d = -1, d \in D; b_o = 0; b_a = 0, a \in A$
          Links to label   : $\bar{A} = \emptyset$
          Dummy arc        : $\bar{a} = (\bar{\imath}, \bar{\jmath}, \bar{m})$ with $\bar{\imath} = 0, \bar{\jmath} = o, \bar{m} = m(p); u_{\bar{a}} = 0$
          Go to step 4.
Step 1 -  Choice of arc to label
          if $\bar{A} = \emptyset$ then STOP.
          Choose $\bar{a} = (\bar{\imath}, \bar{\jmath}, \bar{m})$ of $A$ such that $u_{\bar{a}} \leq u_a$ for all $a \in \bar{A}$.
          Arc $\bar{a}$ receives a parmanent label: $\bar{A} = \bar{A} - \{\bar{a}\}$.
          If $\bar{\jmath}$ is a transfer node, go to step 3; if $\bar{\jmath}$ is a regular
          node, go to step 4; otherwise continue.
Step 2 -  Test of "head" node $\bar{\jmath}$ (for destination node)
          If $u_{\bar{a}} \leq u_{\bar{\jmath}}$ then $u_{\bar{\jmath}} = u_{\bar{a}}; b_{\bar{\jmath}} = \bar{a}$.
          Return to step 1.
Step 3 -  Scan of successors with transfers
          For each $a = (i, j, m)$ such that $i = \bar{\jmath}$ and $m \in m(p)$ do:
             if there is a transfer $t = (\bar{a}, a)$ do:
                if $u_{\bar{a}} + c_t + c_a < u_a$ then $u_a = u_{\bar{a}} + c_t + c_a; b - a = \bar{a}; \bar{A} = \bar{A} \cup \{a\}$
     otherwise, if $m = \bar{m}$ do:
        if $u_{\bar{a}} + c_a < u_a$ then $u_a = u_{\bar{a}} + c_a; b_a = \bar{a}; \bar{A} = \bar{A} \cup \{a\}$
  Return to step 1.
Step 4 -  Scan of successors without transfers
          For each $a = (i, j, m)$ such that $i = \bar{\jmath}$ and $m = \bar{m}$ do:
            if $u_{\bar{a}} + c_a < u_a$ then $u_a = u_{\bar{a}} + c_a; b_a = \bar{a}; \bar{A} = \bar{A} \cup \{a\}$
          Return to step 1.
```

Several remarks concerning this shortest path algorithm are in order. Normally, each arc corresponds to only one mode. The dummy (connector) arc that is incident to the origin node o, has as its mode the set $m(p)$ of permitted modes. This is done so that the first time that step 4 is carried out, all the arcs that leave o belong to the subnetwork. (This may be easily implemented by using an appropriate bit pattern to represent the permitted modes on a link and by using only one word in the computer implemantation of the algorithm.) Thus, the first time that step 4 is executed, the test $m = \bar{m}$ is $m \in \bar{m} = m(p)$. In step 3, the arcs are scanned in increasing order; therefore it is advantageous that the transfers of the type $(\bar{a}, .)$ are sorted in increasing order as well. The arcs that belong to the "forward star" of node i are sorted by mode in the arc table; thus an efficient test may preclude the scan of all the successor links in step 4. The set $\bar{A}$ of labeled links is kept in the form of a binary heap, which makes it trivial to find the link $\bar{a}$ that is the next to be labelled.

We examine next the complexity of this shortest path algorithm. Table 1 shows the complexity of every step of the algorithm and the number of times that it is executed.

The complexity of the entire algorithm for one origin node is then $O(n_A \ln(n_A) + \Delta m(p) n_N)$, where Δ is the average number of nodes connected to the current node by arcs leaving the node. Since the first term dominates the second term in the parentheses, the complexity is then $O(n_A \ln(n_A))$. The shortest path algorithm must be applied for each origin and for each subset of modes $m(p) \in M(p)$. Then, if n_p is the number of products, $|O|$ is the number of the origin nodes and $|M(p)|$ is the number of mode subsets in a given problem, the complexity of the shortest path computations at each iteration of the solution algorithm proposed in section 3 is $O(n_p \ |M(p)| \ |O| \ n_A \ln(n_A))$.

step	complexity	♯ times
0	$O(n_N + n_A)$	1
1	$O(\ln(n_A))$	n_A
2	$O(1)$	$n_D \leq n_N$
3	$O(\Delta m(p))$	n_T
4	$O(\Delta m(p))$	$n_N - n_T$

Table 1 Complexity of the shortest path algorithm

3.7 COMPUTATIONAL RESULTS

We first present in this section computational results obtained on a small network, which permits the comparison of the Gauss-Seidel - Linear Approximation algorithm (a) with the Linear Approximation Algorithm (b).

The test network consists of 5 origins, 5 destinations, 19 regular nodes, 116 links, and 226 transfers. Seven modes are permitted on the links of this network. The following test problems were contructed :

1. A single product is transported on the network and all modes are permitted. There is one origin/destination matrix. In this case (a) and (b) are equivalent as can be seen in Figure 6. The slight difference in the computational time is that in (a) the gap is recomputed at the end of a major cycle, whereas in (b) this gap is available automatically.
2. Three products are assigned to the network, where two products are permitted on all modes and the third is restricted to a subset of three modes. The demands

of each of the three products are given by one matrix. In this case algorithm (a) is slighty more efficient than (b), as evidenced by Figure 7.

3. Five products are assigned to the network but only two may be transported by all the modes. The demand for the first product is given by three origin/destination matrices, each for different subsets of modes. Similarly, the demand for the second product is divided into two matrices. Figure 8 shows that the performance of the algorithm (a) is very good.

4. The same five products of the previous test problem and the same orgin/destination matrices are assigned to the network with the difference that all the modes are permitted. In addition, the generalized cost is the weighted sum of three costs (operation, delay, and energy), whereas in the previous problems, only the delay cost was considered. This has, as a consequence, the tripling of the evaluation time of the unit cost functions, which, for problems of this size, represent the largest part of the computational time. The results are shown in Figure 9.

We note that in all cases where there is more than one product, algorithm (a) is more efficient than algorithm (b) both in terms of number of iterations and computational time (we note here that these results were obtained on an MC 68000 microcomputer not equipped with a floating point accelerator). Since algorithm (a) requires less central memory (RAM) than algorithm (b), it is well suited for problems of larger size.

We conclude this section by giving the computational results obtained on the network that represents the national freight transportation network of Brazil. This network consists of 211 origins and destinations, 1243 regular nodes, 4957 links and 5718 transfers. Six products (cement and steel products, iron ore, fertilizer and coal, soya oil, soya grain, and all others) are assigned and 10 modes are permitted on the links of this network. Only the delay cost was considered. The test was carried out on a SUN/3 work station (MC 68020 equipped with DC 68881 floating point accelerator operating at 16.7 Mhz).

Figures 10 and 11 show the decrease of the objective function vs the number of iterations and the total computation time respectively. Figures 12 and 13 show the decrease of the relative gap vs the number of iterations and the total computation time respectively.

While the computational time per iteration appears to be high, approximately 80% of the time of an iteration is spent on computing the marginal costs and the gap. This is due to our choice of making the code "user friendly" and require the user to enter the cost functions directly and not impose on time the calculation of the analytical form of the cost functions.

3.8 CONCLUSIONS

We have developed a multi-mode multi-product network assignment model for strategic planning of freight flows, that permits the modeling of national and regional freight transportation networks. The multiproduct aspects of the model are exploited in the solution procedure, which is a Gauss-Seidel Linear Approximation Method. The computational results obtained demonstrate that the method is applicable to large multi-modal transportation networks.

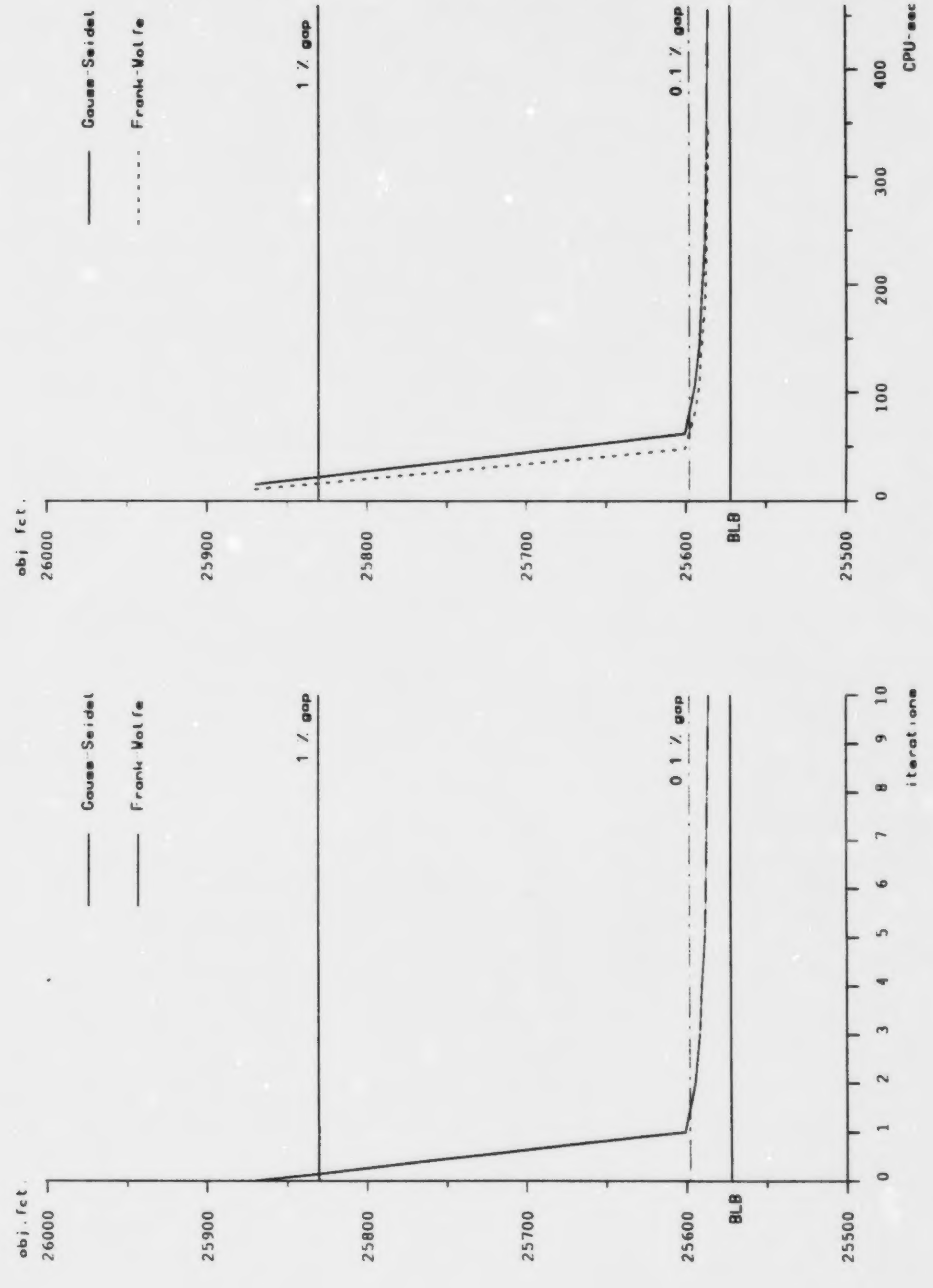


Figure 5: Problem 1

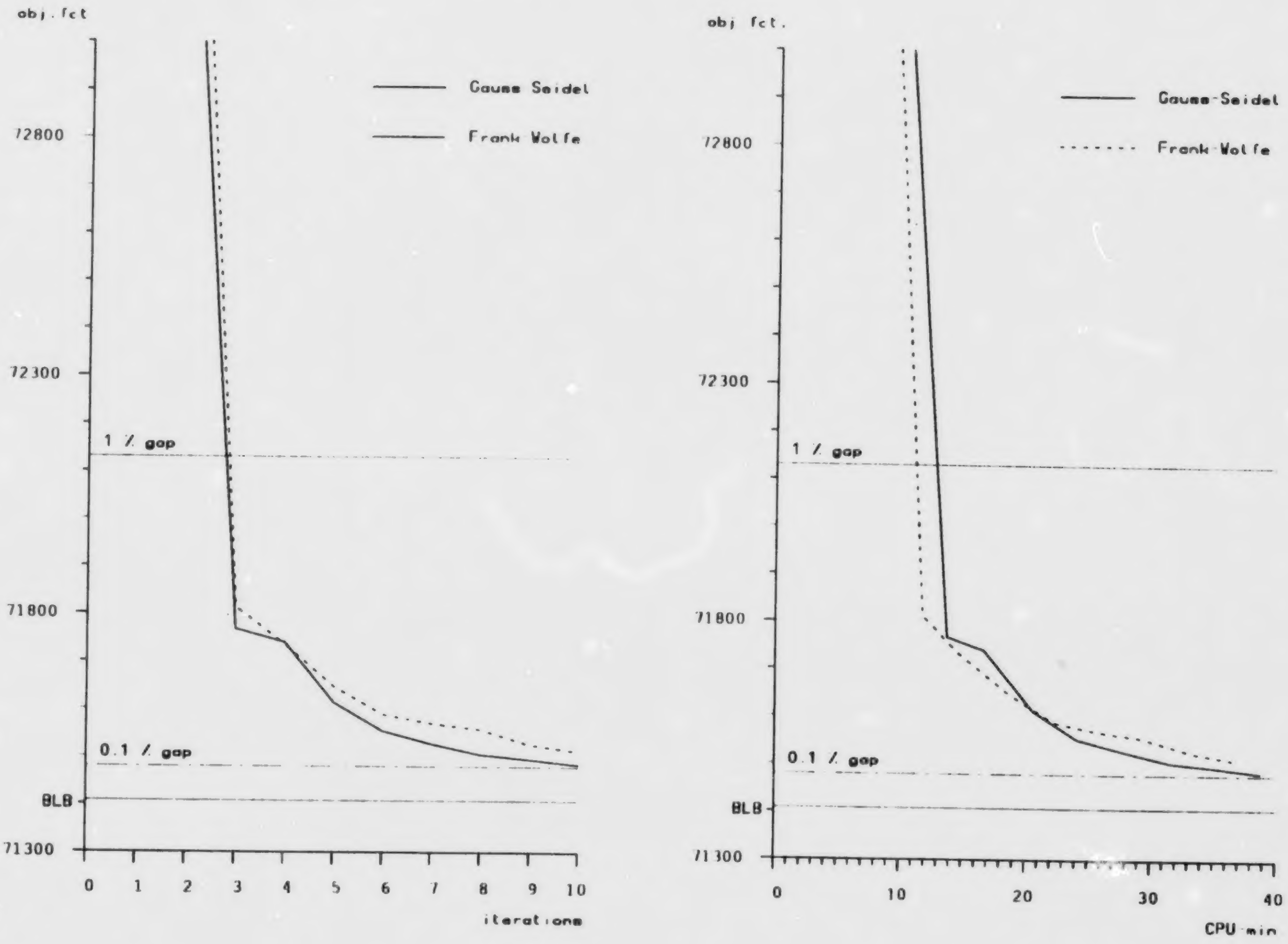


Figure 6: Problem 2

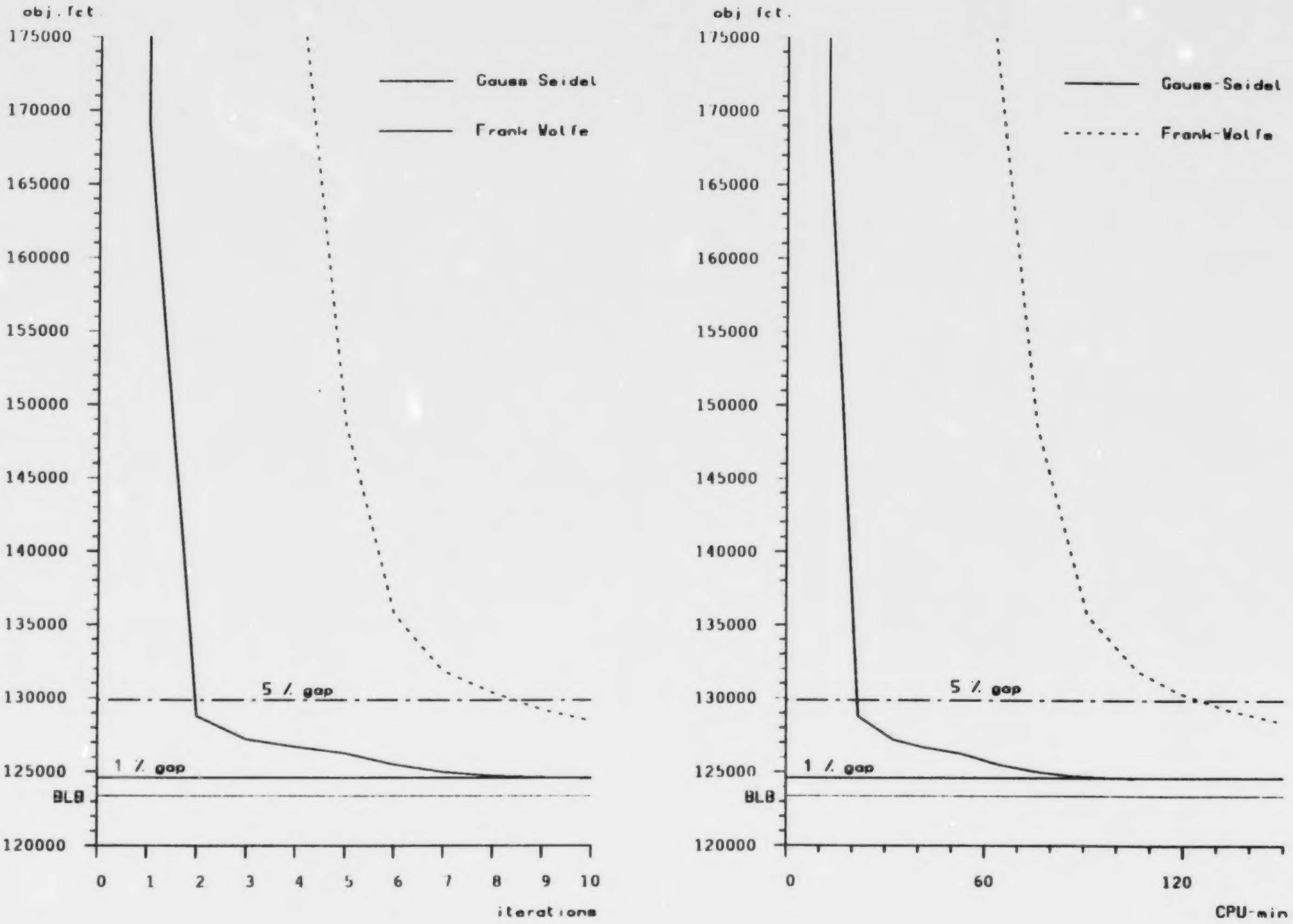


Figure 7: Problem 3

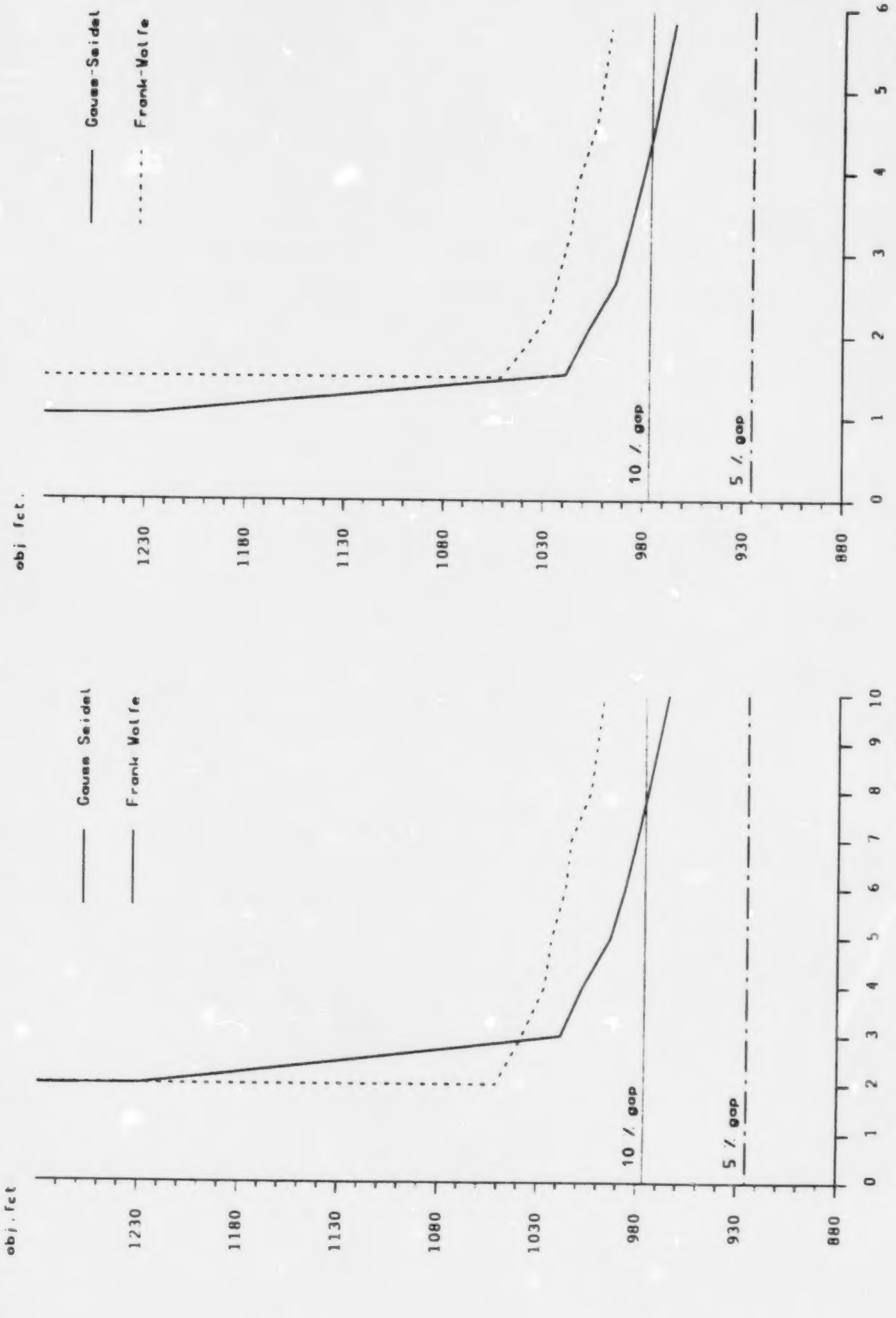


Figure 8: Problem 4

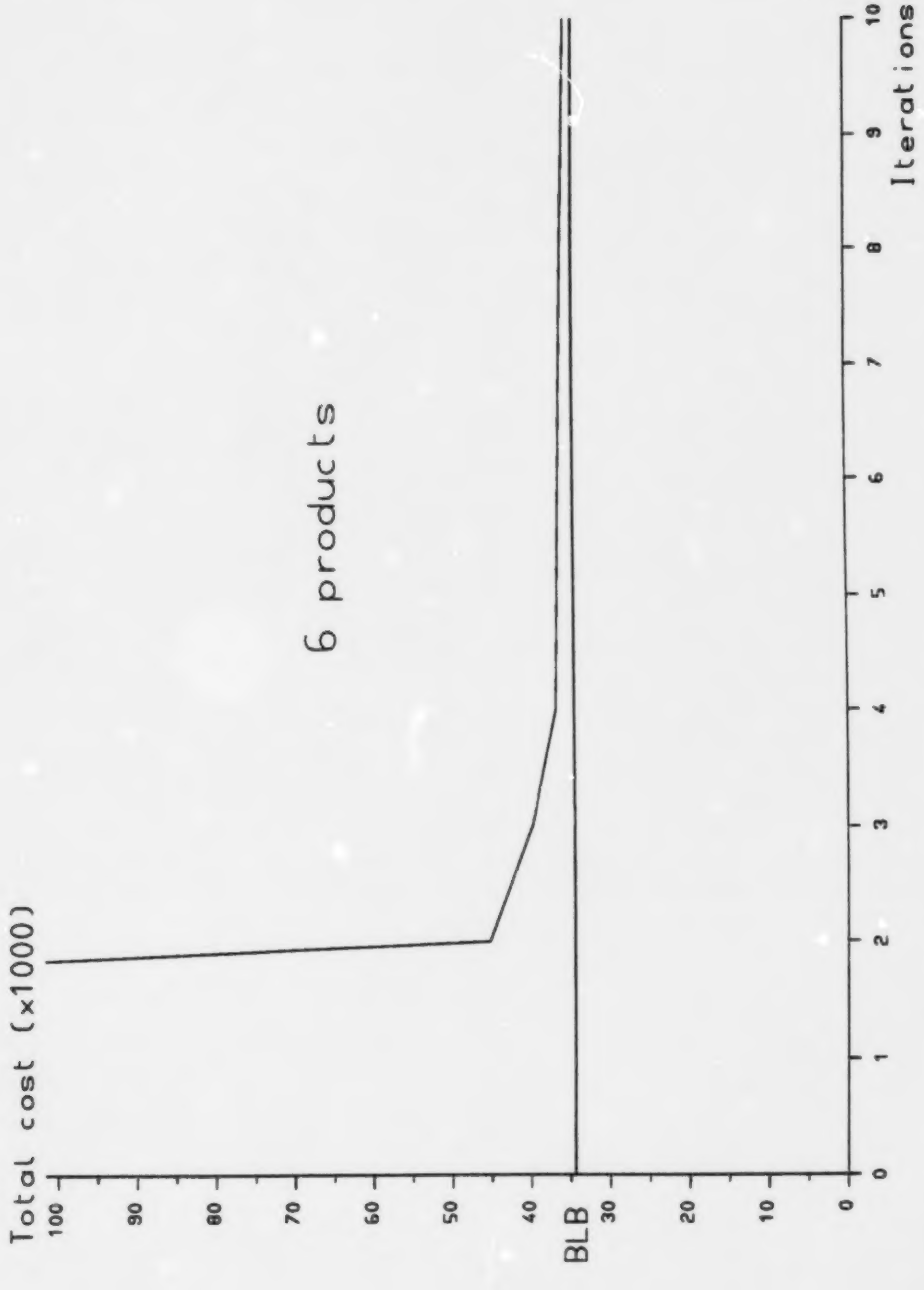

6 products
Total cost (×1000)
100
90
80
70
60
50
40
BLB
30
20
10
0
0
1
2
3
4
5
6
7
8
9
10
Iterations

Figure 9

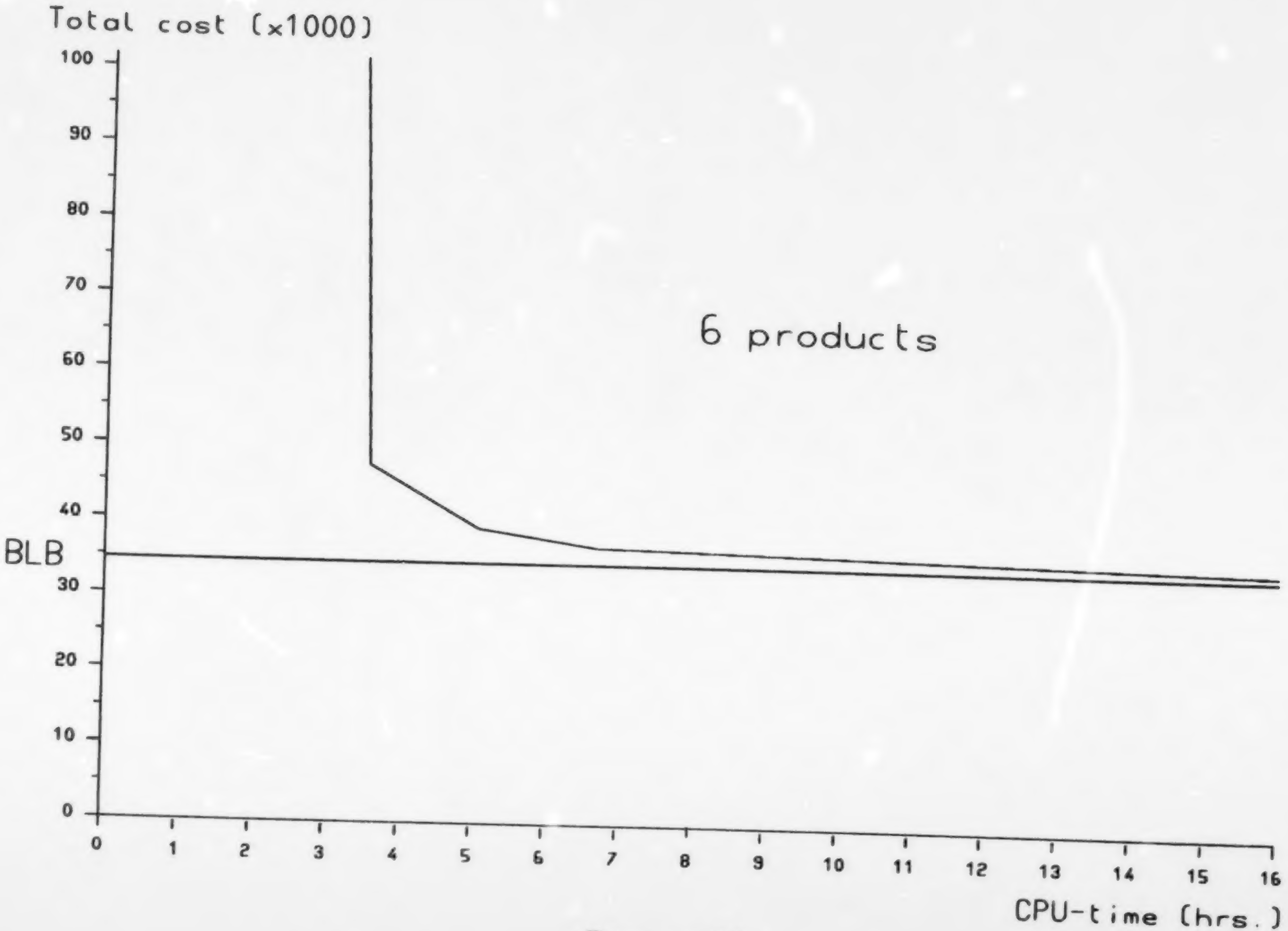

Total cost (x1000)
100
90
80
70
60
50
40
BLB
30
20
10
0
6 products
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
CPU-time (hrs.)

Figure 10

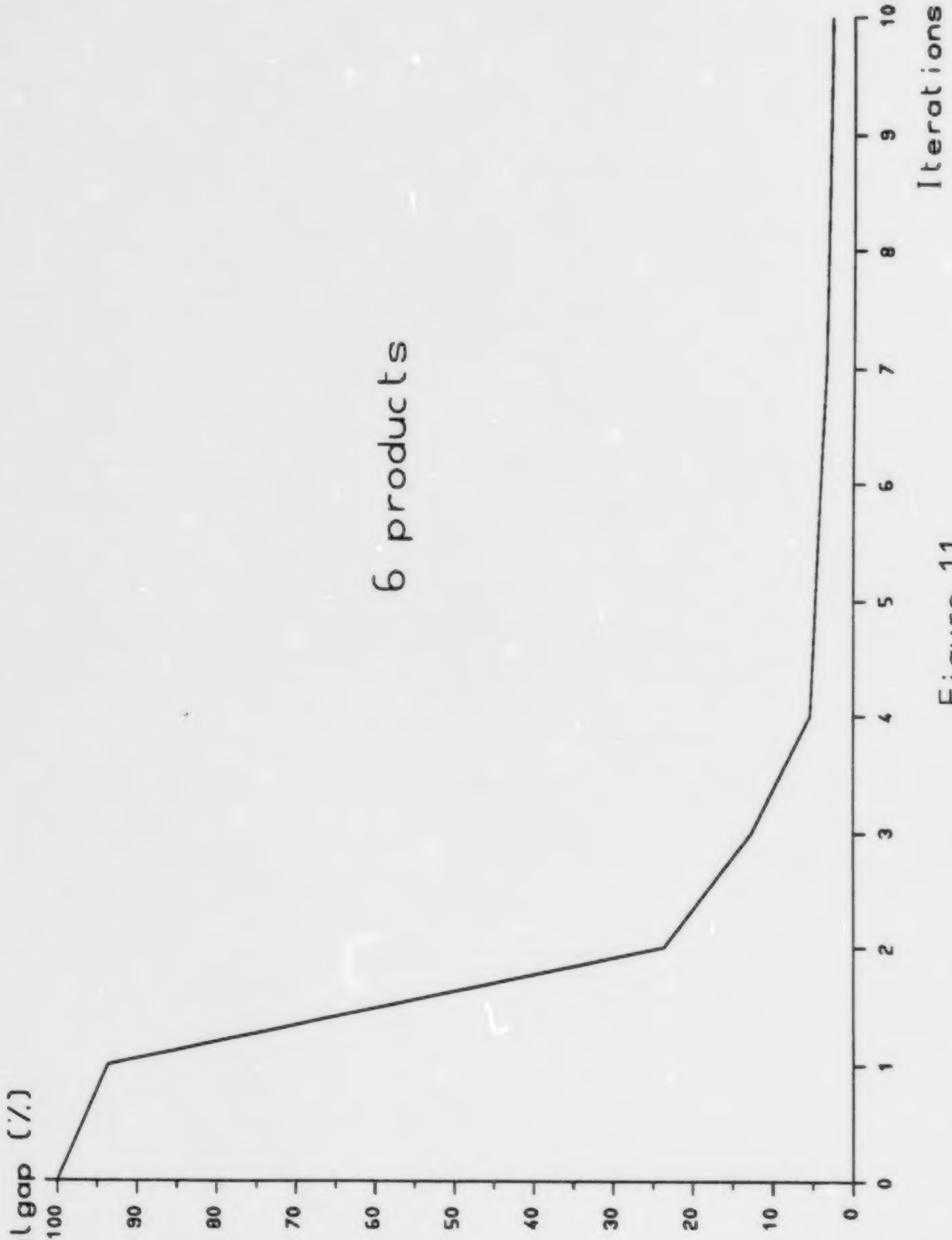

6 products
Relgap (%)
100
90
80
70
60
50
40
30
20
10
0
0
1
2
3
4
5
6
7
8
9
10
Iterations

Figure 11

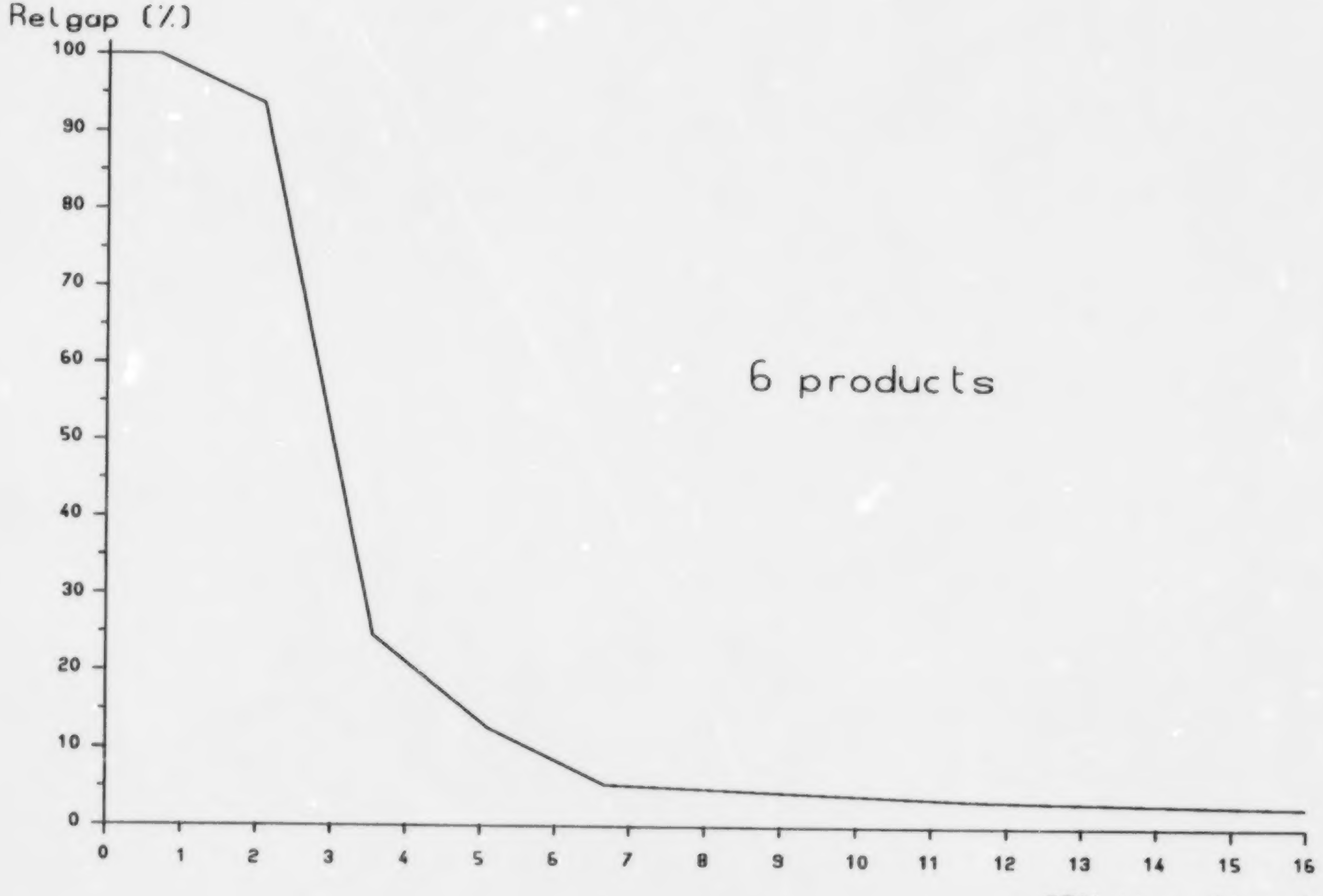

Relgap (%)
100
90
80
70
60
50
40
30
20
10
0
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
CPU-time (hrs.)
6 products

Figure 12

3.9 BIBLIOGRAPHY

Bronzini, M.S. (1980) "*Evolution of a Multimodal Freight Transportation Network Model*", **Proceedings of the 21st Annual Meeting, Transportation Research Forum**, 465-485.

Casaletto, J.J., Rice, J.R. (1978) "*Smoothing and Estimating Derivatives of Equispaced Data*", **Applicable Analysis** 8,143-152.

Florian, M., Los, M. (1982) "*A New Look at Static Price Equilibrium Models*", **Regional Science and Urban Economics** 12,579-597.

Florian, M. (1984) "*An Introduction to Network Models Used in Transportation Planning*", **Transportation Planning Models**, M. Florian (ed.), North-Holland, Amsterdam, 137-152.

Florian, M. (1986) "*Nonlinear Cost Network Models in Transportation Analysis*", **Mathematical Programming Study** 26,167-196.

Frank, M., Wolfe, P. (1956) "*An Algorithm for Quadratic Programming*", **Naval Research Logistics Quarterly** 3,95-110.

Friesz, T.L., Tobin, R.L., Harker, P.T. (1983) "*Predictive Intercity Freight Network Models*", **Transportation Research** 17A,409-417.

Friesz, T.L., Gottfried, J.A., Morlok, E.K. (1986) "*A Sequential Shipper-Carrier Network Model for Predicting Freight Flows*", **Transportation Science** 20,80-91.

Guélat, J. (1986) "*Une procédure de résolution du problème d'affectation multimode-multiproduct par décomposition sur un espace produit*", publication #488, Centre de recherche sur les transports, Université de Montréal.

Guélat, J., Florian, M., Crainic, T.G. (1987), "*A Multimode Multiproduct Network Assignment Model for Strategic Planning of Freight Flow*", publication # 549, Centre de Recherche sur les transports, Université de Montréal, to appear **Transportation Science**.

Jones, P.S., Sharp, G.P. (1977) "*Multi-Mode Intercity Freight Transportation Planning for Underdeveloped Regions*", **Proceedings of the 18th Annual Meeting, Transportation Research Forum** (523-531).

Kellison, S.G. (1975) "*Fundamentals of Numerical Analysis*", Richard D. Irwin Ink., Homewood, Illinois.

Kresge, D.T., Roberts, P.O. (1971) "*System Analysis and Simulation Models*", in *Techniques of Transport Planning*, Meyers Jr. (ed), vol. 2, Brookings Institute, Washington, D.C.

McGinnis, L.F., Sharp, G.P., Yu, D.H.C. (1981) "*Procedures for Multi-State, Multi-Mode Analysis: Vol. IV* ", Transportation Modeling and Analysis, U.S., D.O.T. report no. DOT-OST-80050-17/V.N.

Potts, R.B., Oliver, R.M. (1972) "*Flows in Transportation Networks*", Academic Press NY.

Samuelson, P.A. (1952) "*Spatial Price Equilibrium and Linear Programming*", **American Economic Review** 42,283-303.

Sharp, G.P. (1979) "*A Multi-Commodity Intermodal Transportation Model* ", **Proceedings of the 20th Annual Meeting Transportation Research Forum**, 399-407.

Spiess, H. (1984) "*Contributions à la théorie et aux outils de planification des réseaux de transports urbains*", Ph.D. thesis, Département d'informatique et de recherche opérationnelle, publication #382, Centre de recherche sur les transports, Université de Montréal.

Takayama, T., Judge, G.G. (1964) "*Equilibrium among Spatially Separated Markets: A Reformulation*", **Econometrica** 32,510-524.

Takayama, T., Judge, G.G. (1970) "*Alternative Spatial Price Equilibrium Models*", **Journal of Regional Science** 10,1-12.

Zangwill, W.I. (1969) "*Nonlinear Programming - A Unified Approach*", Prentice Hall, Englewood Cliffs, NJ.

3.9 BIBLIOGRAPHY

Bronzini, M.S. (1980) *Evolution of a Multimodal Freight Transportation Network Model*, **Proceedings of the 21st Annual Meeting, Transportation Research Forum**, 475-485.

Casaletto, J.J., Rice, J.R. (1978) *Smoothing and Estimating Derivatives of Equispaced Data*, **Applicable Analysis** 8,143-152.

Florian, M., Los, M. (1982) *A New Look at Static Price Equilibrium Models*, **Regional Science and Urban Economics** 12,579-597.

Florian, M. (1984) *An Introduction to Network Models Used in Transportation Planning*, **Transportation Planning Models**, M. Florian (ed.), North-Holland, Amsterdam, 137-152.

Florian, M. (1986) *Nonlinear Cost Network Models in Transportation Analysis*, **Mathematical Programming Study** 26,167-196.

Frank, M., Wolfe, P. (1956) *An Algorithm for Quadratic Programming*. **Naval Research Logistics Quarterly** 3,95-110.

Friesz, T.L., Tobin, R.L., Harker, P.T. (1983) *Predictive Intercity Freight Network Models*, **Transportation Research** 17A,409-417.

Friesz, T.L., Gottfried, J.A., Morlok, E.K. (1986) *A Sequential Shipper-Carrier Network Model for Predicting Freight Flows*, **Transportation Science** 20,80-91.

Guélat, J. (1986) *Une procédure de résolution du problème d'affectation multimode-multiproduct par décomposition sur un espace produit*, publication #488, Centre de recherche sur les transports, Université de Montréal.

Guélat, J., Florian, M., Crainic, T.G. (1987), *A Multimode Multiproduct Network Assignment Model for Strategic Planning of Freight Flow*, publication # 549, Centre de Recherche sur les transports, Université de Montréal, to appear **Transportation Science**.

Jones, P.S., Sharp, G.P. (1979) *Multi-Mode Intercity Freight Transportation Planning for Underdeveloped Regions*, **Proceedings of the 20th Annual Meeting, Transportation Research Forum** (523-531).

Kellison, S.G. (1975) *Fundamentals of Numerical Analysis*, Richard D. Irwin Ink., Homewood, Illinois.

Kresge, D.T., Roberts, P.O. (1971) *System Analysis and Simulation Models*, in *Techniques of Transport Planning*, Meyers Jr. (ed), vol. 2, Brookings Institute, Washington, D.C.

McGinnis, L.F., Sharp, G.P., Yu, D.H.C. (1981) *Procedures for Multi-State, Multi-Mode Analysis: Vol. IV*, Transportation Modeling and Analysis, U.S., D.O.T. report no. DOT-OST-80050-17/V.N.

Potts, R.B., Oliver, R.M. (1972) *Flows in Transportation Networks*, Academic Press NY.

Samuelson, P.A. (1952) *Spatial Price Equilibrium and Linear Programming*, **American Economic Review** 42,283-303.

Sharp, G.P. (1979) *A Multi-Commodity Intermodal Transportation Model*, **Proceedings of the 20th Annual Meeting Transportation Research Forum**, 399-407.

Spiess, H. (1984) *Contributions à la théorie et aux outils de planification des réseaux de transports urbains*, Ph.D. thesis, Département d'informatique et de recherche opérationnelle, publication #382, Centre de recherche sur les transports, Université de Montréal.

Takayama, T., Judge, G.G. (1964) *Equilibrium among Spatially Separated Markets: A Reformulation*, **Econometrica** 32,510-524.

Takayama, T., Judge, G.G. (1970) *Alternative Spatial Price Equilibrium Models*, **Journal of Regional Science** 10,1-12.

Zangwill, W.I. (1969) *Nonlinear Programming - A Unified Approach*, Prentice Hall, Englewood Cliffs, NJ.

CHAPTER FOUR

STRATEGIC RAIL MODELING

Rail is an important part of any serious national freight transportation study. Consequently, a significant amount of the understanding and methodological effort necessary to build a comprehensive model of a national transportation system has to be dedicated to an adequate representation of the rail mode and its interactions with the other components of the transportation system. Several reasons explain this fact.

Rail usually transports large quantities of freight on relatively long distances and is perceived as a most energetically efficient transportation mode in these cases. Therefore, it is an important economic activity that involves infrastructure and equipment representing significantly large capitalisations and that employs, either directly or undirectly, a large work force. Modifications to rail infrastructure generally require large investments and long periods of time both for their planning and realisation.

Like other transportation modes, transport by rail is usually regulated by one or several governmental agencies. Often, rail companies are either state owned or state controled. Consequently, rail transportation is often viewed, especially in developing countries, as a privileged economic and social lever to boost national unity and integrity, to control energy costs, to open up new regions to population and economic development, to bring products for export from the producing regions to ports and to transport imported goods to the consumption centers of the country, etc. This view is equally often shared by the international agencies which usually supply the funds for projects in developing countries that require large investments.

For STAN, we developed a modeling of the rail freight transportation system, its components, operations and goals, that is compatible with both the strategic objectives of the planning system and the representation of the other transportation modes that compose a national multimode transportation system.

The objective of this chapter is to present and analyse this strategic model of a national freight rail transportation system. We also examine how the rail model is integrated into STAN, as well as how the system may be used to model a number of realistic scenarios.

The report is organized as follows. Section one presents a brief review of the components, organization and operations of a rail freight transportation system, as well as a brief survey of some previous rail strategic modeling efforts. Section two consists of a brief remainder of STAN's modeling concepts and framework and of their impact on the representation of the rail mode. Section three is dedicated to the detailed description of the rail network model and the formulation of the cost and delay functions. In Section four we briefly discuss the empty car distribution problem in the context of strategic planning. We conclude with the presentation of possible utilization modes of the rail model.

4.1 RAIL FREIGHT TRANSPORTATION

The physical infrastructure of any rail system consists of a complex network made up of a number of large terminals *classification yards* , numerous small stations and junction points interconnected by main and secondary lines. At any given moment, a large number of rail cars, either empty or loaded with a specific commodity, are in the network, and each one may have a particular objective, independent of all the others. The Canadian National Railways' network, for example, has over 5000 yards and stations and serves 22,000 operated mainline miles of track. On this network, one finds more than 100,000 CNR and foreign cars (CNR, 1983). Similarly, some 75,000 cars move on the 30,000 kilometers of track of the Brazilian railway network, which comprises three gauges (narrow, metric and large, plus a mixed metric-large gauge), two traction modes (diesel and electric plus a mixed mode) and is operated by three main companies besides a number of small ones.

Cars do not move independently, however, but in formations called blocks, which in turn are grouped into convoys called *trains*. Blocks are groups of cars, with possibly different final destinations, arbitrarily considered, for all operational purposes, as a single unit from the yard where the block is assembled till its destination yard where its component cars may be reclassified and put in a new block. One or more blocks, plus the necesssary engines, make up a train.

All station to station movements in a rail network are performed by trains. Trains travel following predefined routes (and schedules, eventually) at specified frequencies over a planning period. The route, frequency and other operational characteristics, such as train type, speed, priority, etc., define the *train service*. On a medium to long term planning horizon, the train service network specifies all possible movements throughout the network and thus defines the operating cost structure of the system.

Characteristically, rail vehicles are captive to the infrastructure: cars and engines move exclusivelly on rail tracks. This captivity and the fact that in most rail systems several different types of train services (e.g. passenger, fast freight, unit, normal freight, etc.) are operated concomitantly, sometimes on single track rail lines, generate the train meetings and overtakes on the tracks of the network which are so characteristic of the rail mode. Consequently, other than the usual physical *capacities* of the links, in analysing rail transportation one has also to pay considerable attention to the *over-the-line delays* which represent, for a specific train type, the travel time over the link, given the link characteristics and the *train congestion*, generated by the total train traffic in both directions on the link.

This organization and its successive aggregations, summarized in Figure 1, are characteristic to the rail mode and, when properly planned, generate economies of scale resulting in increased efficiency. They are also responsible, however, for more complex yard operations, more intricate cost and delay formulae and additional planning problems.

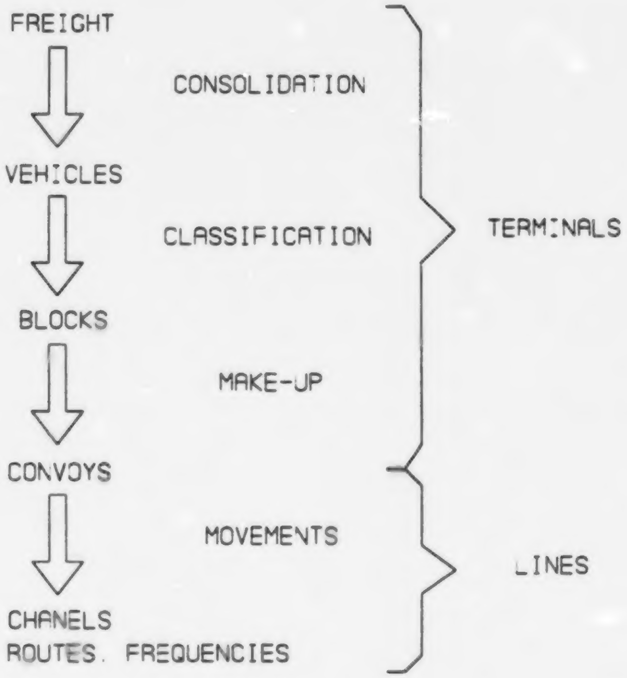

FREIGHT
CONSOLIDATION
VEHICLES
CLASSIFICATION
TERMINALS
BLOCKS
MAKE-UP
CONVOYS
MOVEMENTS
LINES
CHANELS
ROUTES. FREQUENCIES

Figure 1 Rail Orgnization

To control and plan the operations of such a generally vast and complex system, rail companies establish a number of rules and policies at all levels of decision making in the company. Thus, operational policies deal dynamically and in great detail with local operations: train time-tables, track priority rules, yard dispatching policies, engine and empty car scheduling and so on. These policies generally follow the guidelines established at the tactical level of decision-making, where general, network-wide operating strategies are generated. The resulting ***transportation plan*** specifies general yard car classification and train formation policies, the design of the service network (the train services to run, their routes and frequencies) and the routing of traffic through the physical and service networks, to satisfy global economic and service criteria. Strategic decisions are concerned with the long term evolution of the company: major investments, important modifications to the physical infrastructure or to the operating policies, etc. Often the possible impact of strategic decisions is evaluated by generating and analysing the corresponding transportation plan. Assad (1980a, 1980b), Crainic (1987a, 1987b), Dejax and Crainic (1987) present model formulations (mostly from the operations research field) aimed at these problems.

The rail mode is present in most models of national/regional multimode transportation systems. It has also received quite a great deal of attention as a single mode multicarrier problem (Crainic, 1987a). However, none of these modeling approaches answers our aim. The Harvard Model (Kresge and Roberts, 1971), is the first significant predictive freight network model which was conceived for the national planning of a developing country. It has only "modal" routes as links (no representation of the physical network) and congestion phenomena are not considered. The Multi-State Transportation Corridor Model (McGinnis et al., 1981, Jones and Sharp, 1979, Sharp, 1979) considers an explicit multimodal network but, costs are linear, congestion is ignored and backhauling is not considered. Congestion is also ignored in the Transportation Network Model (Bronzini, 1980). The Freight Network Equilibrium Model (Friesz, Gottfried and Morlok, 1986) does include nonlinear costs and delay functions that model congestion phenomena; but, it is a sequential model using two network representations: an aggregated skeletal user-perceived network where product flows are determined by mode and carrier, and separate networks for each carrier acting alone to minimize its costs, akin to a tactical planning perspective. The single mode rail formulations (Kornhauser et al., 1979, Lansdowne, 1981, Bronzini and Sherman, 1983, etc.), generally consider more detailed network and operational representations, even if backhauling is not usually considered. However, there are no simple ways of integrating them into more general multimode models.

Our goal is to achieve an aggregate, *strategic* , modeling of the transportation of freight by rail that may, on one hand, adequately reflect the economics and spatial-temporal relations typical to the rail mode. We have to acknowledge, therefore, the importance of the train movements and yard operations, especially car classification, that define a large part of the system's operating costs and generate congestion, and thus delays, in the yards and on the links of the network. On the other hand, our formulation has also to respect the conventions and general guidelines of the general model for the strategic planning of a multimode multicommodity national freight transportation system on which STAN is based.

4.2 THE MODEL

The network optimization model that is used to simulate network flows in STAN is a non-linear multimode multicommodity assignment formulation that minimizes the *total generalized system cost* while satisfying the usual flow conservation constraints:

$$\mathit{Minimize}\ \sum_{p\in P}\left[\sum_{a\in A} s_a^p(v) * v_a^p + \sum_{t\in T} s_t^p(v) * v_t^p\right]$$

Subject to

$$\sum_{k\in K_{od}^{m(p)}} h_k = g_{od}^{m(p)} \qquad o\in O, d\in D, m(p)\in M(p), p\in P$$

$$h_k \geq 0 \qquad k\in K_{od}^{m(p)}, o\in O, d\in D, m(p)\in M(p), p\in P$$

where,

N : set of nodes;

L : set of links;

T : set of transfers;

O : set of nodes where demand originates;

D : set of nodes where demand terminates;

M : set of modes;

$m(p)$: set of permitted modes for produit P

$K_{od}^{m(p)}$: set of multimodal paths joining origin o to destination d for product p by using only nodes of $m(p)\in M(p)$;

$g_{od}^{m(p)}$: demand of product p to be moved from origin o to destination d;

h_k : flow on path k, $k_{od}^{m(p)}$;

v_a^p : flow of product p on link a

$$v_a^p = \sum_{k\in K^p} \delta_{ak} * h_k \qquad a\in A, p\in P;$$

δ_{ak} : 1 if link a belongs to path k; 0 otherwise;

v_t^p : flow of product p on transfer t

$$v_t^p = \sum_{k\in K^p} \delta_{tk} * h_k \qquad t\in T, p\in P;$$

δ_{tk} : 1 if transfer t belongs to path k; 0 otherwise;

K^p : set of all paths for product p by using the permitted modes $M(P)$;

s_a^p : generalized cost on link a for product p;

s_t^p : generalized cost on transfer t for product p.

According to the definition of the mode set M, this formulation represents either the entire national/regional multimode transportation system, or a part of it, a subsystem of specified transportation modes. It is thus the mathematical formulation of the national freight rail transportation system that we use.

A detailed description of the general STAN model may be found in the previous chapters. Here, we only highlight some of the main features of the formulation, that are relevant for the modeling of the rail mode:

- All traffic (i.e., all flow variables) are mesured in tons only;
- There are no explicit capacity constraints on the links, nodes or transfers of the network;
- Parallel links are allowed in the network, each link being used exclusively by a specific mode;
- Transfers may be defined at a node between specific modes;
- Total generalized system cost is the sum of the individual link and transfer generalized costs;
- The global objective, i.e. the total generalized system cost, is assumed to be a convex differentiable function, as required by the solution method described in chapter 3.

This modeling framework provides both an adequate representation of large multimode multiproduct transportation systems for strategic planning purposes and a mathematical structure well suited for efficient solution methods even when large problems are solved on small computers. It also imposes limitations in the representation of the individual transportation modes, especially rail. We have, in spite of this, achieved a model of rail freight transportation that reconciles the requirements of strategic planning with the complexity of rail operations. The details of the model related to the rail mode is the object of this chapter.

4.3 RAIL MODELING

The requirements of a national model for the evaluation of strategic decisions imply that a number of typical rail elements may not be explicitly described on the network. In particular, path information, from origins to destinations, is not available in the model, implying that train services and frequencies cannot appear in the formulation. Hence, approximations are used.

The network representation, on the other hand, has to be rather detailed and specific, since an important class of scenarios considered in STAN reflect modifications in the infrastructure and their impact on the utilization of the various transportation modes. The network model should therefore be able to reflect adequately the interaction between the physical characteristics of the infrastructure and the performance of the transportation system: traffic distribution and mode utilization, congestion and bottlenecks, etc.

Our modeling framework aims at satisfying these objectives. We first present the network representation, then the flow and cost functions and, last, the formulation of the delay functions.

4.3.1 The modeling of the rail network

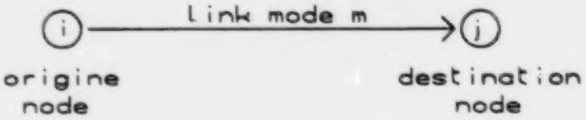
The rail infrastructure is represented by a network, $G = (N, A, T)$, composed of nodes, links and transfers.

The set of *nodes* includes centroids and regular nodes. Regular nodes represent physical locations on the network: yards, stations, junctions, intermodal transfer facilities, etc. *Centroids* are dummy nodes that correspond to regional subdivisions; all the demand for transportation, that either originates from or terminates at a particular region is "concentrated" at the associated centroid. Centroids make up the sets O and D. No regular node may be the origin or the destination of any traffic.

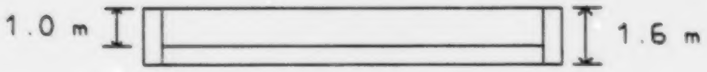
Links represent the rail tracks that connect two points in N. Links have a number of physical characteristics, such as length, capacity, etc. We further examine the notion of link capacity in Section 4.3.3. Product volumes, v_a^p for $p \in P$, and the operating cost, delay and energy consumption functions necessary to compute the generalized cost are associated with each link a .

The network is composed of directed links with one mode allowed on each link, as illustrated in Figure 2a. When more than one mode serves two adjacent nodes, parallel links may be defined between these nodes. Some, or all, of these parallel links may share the same infrastructure or capacity.

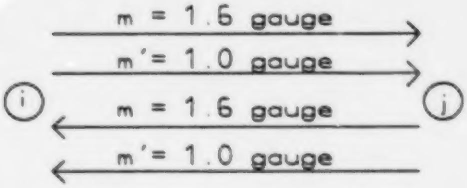
These features allow, for example, the representation of different rail companies, track gauges or, under some additional hypotheses and limitations, various classes of transportation services as different transportation modes. Figure 2 illustrates the richness provided by this modeling approach: a single track with double gauge (physically, three rails are installed and trains of both gauges travel in both directions over the line - see Figure 2b) is represented as four links: two (one for each direction of travel) links for each gauge defined as a separate mode. Note that, in this example, the four links share the same capacity.



a) link l = (i.m.j)



b) single-track double-gauge line



c) the four corresponding links

Figure 2 Rail Line Modelization

Connectors are a particular group of links that "connect" the centroids to the rest of the network. Connectors have, therefore, all the characteristics and properties of regular links and are used to model the access to the transportation infrastructure. For instance, one may use connectors to model the situation where some specific products enter or leave a given region via some particular stations. In the example illustrated in Figure 3, products may leave the region-centroid **C** either via node **i** or node **j** and products may enter the region either via node **i** or node **k** . Node **h** is neither an initial nor a terminal node for any traffic movement. Furthermore, by adroitly adjusting the cost fuctions (e.g. infinite transportation costs on the C

- j connector), one may specify that a particular group of products leave the region exclusively through, say, node i .

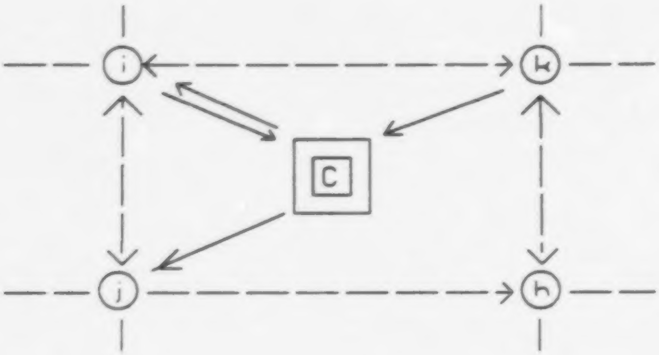
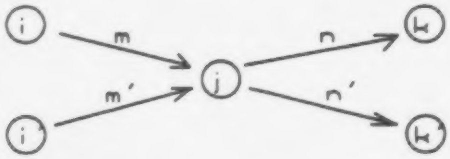


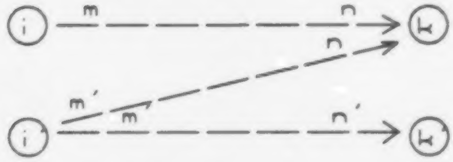
Figure 3 Connectors

Transfers represent intermodal movements at nodes, as illustrated in Figure 4. In our model, freight paths may be composed of links with different modes, only if the appropriate transfers have previously been defined at the nodes of the path. In STAN, transfers are represented by a quintuple that identifies the transfer node, the node and mode (hence, the link) where the transfer initiates, as well as the node and mode of destination of the transfer (see Figure 4a). However, it is also useful to refer to the link-like representation of transfers, as illustrated in Figure 4. Functions, of cost, delay and energy consumption, and product flows, v_t^p for $p \in P$, are associated with each transfer.

Transfers may evidently be used to model intermodal transfer facilities between rail and other transportation modes. For the rail mode, transfers may also be used to represent the junction points between the networks of two different rail companies and the transfer operations performed therein, or the transshipment points from one gauge to another in a rail company's network. Transfers could be used to model classification yards but the resulting formulation is not, however, adequate for most situations, as explained below.



a) transfers $t_1 = [i.m.j.n.k]$
$t_2 = [i'.m'.j.n.k]$
$t_3 = [i'.m'.j.n'.k']$



b) transfer representation as links
$t_1 = [i.\ m - n.\ k]$
$t_2 = [i'.\ m' - n.\ k]$
$t_3 = [i'.\ m' - n'.\ k']$

Figure 4 Transfers

Classification yards sort and group cars, dismantle and make-up trains. Yard workloads, costs and delays are usually mesured in terms of how many trains are received, dismantled and sorted, of the number of cars and groups of cars that are handled, of how many blocs and trains are formed for the other yards in the network. Yard delays are particularly sensitive to the traffic intensity and to the yard's specific characteristics and operating policies.

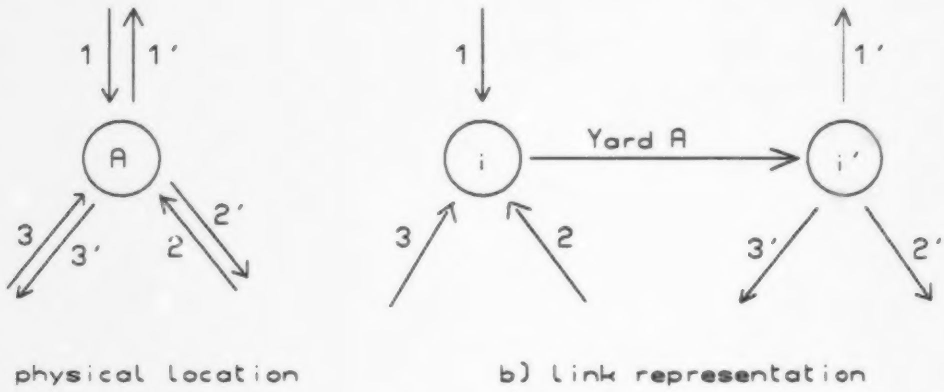
Petersen (1977a,b) presents a classical description of classification yard configurations, operations and performance mesures, as well as a series of models for the main yard operations. Several other authors have built upon these bases and Assad

(1980a) and Crainic (1987b), among others, present recent reviews of this subject. These models generally are network representations (more or less detailed, according to the planning level involved) of either the movements or the sequence of operations performed on the vehicles passing through the yard. Various queueing models are then used to derive the relevant yard performance mesures such as the average classification delay, the average number of cars classified in a time period or the mean total put-through time. This modeling is usually done at a lower level of detail than the one that we consider, generally requiring train or traffic routing information. Such detailed formulations are therefore not compatible with the aggregation level of a strategic model of regional/national multimode transportation systems. We resort then to an aggregate representation of classification yards that approximates the corresponding yard costs and delays using the product flow information exclusively.

The particular cost and delay functions that relate to the yard model are presented in the following sections. Figure 5 illustrates the yard representation. A yard in the physical rail network is usually represented by a node of our network. This node is split, each one of the resulting nodes being the extremity of a link which now represents the classification yard. The origin node receives all the incoming traffic, while the outgoing traffic exits via the destination node. The usual functions associated with links are then employed to approximate the yard throughput performance.

There is one situation where transfers, and their associated functions, could be used to model classification yards and operations. This is the case where only one rail line enters and exits the yard, which occurs when, for example, the yard is situated at the "entry" of a port and all trains have to pass through the yard to be dismantled and their cars be send to particular port terminals.

On this network, transportation activities are induced by the transportation demand. The demand is defined as a number of tons of a given product ($p \in P$), to be moved from an origin point (that is, a centroid in O) to a destination point (i.e., a centroid in D). We want to emphasize the fact that the particularities of the rail mode require the explicit consideration of the demand for empty vehicles. We further examine this aspect of the demand in Section five.



a) physical location

b) link representation

Figure 5 Yard representation

4.3.2 Cost and flow modeling

The unit "costs" that are usually considered in aggregate models of the rail transportation system give, for yards, links and transfers, train and car operating costs in monetary units per train-km and car-km, respectively, train and car delay costs (covering crew costs, equipment and product depreciation, etc.) in monetary units par train-hour and car-hour, respectively, and train energy consumption factors that are dependent upon the link topography. Of course, the cost structure in a real rail system is significantly more complex, yet, for strategic (and even tactical) planning, these unit costs are sufficient to model most situations of interest.

The important fact to note in this context, is that practically all unit costs apply to vehicles and convoys. We have, therefore, to convert our flows of products (in tons) into corresponding numbers of cars and trains. Furthermore, this transformation has to be done locally, at the level of each individual link: car and train flows are not necessarily conserved at nodes. Several approximations have to be made.

We first assume that a *typical car* may be defined for each product $p \in P$ and that each product is exclusively transported by a typical car. This car may be defined, for example, by taking the weighted average (for each of their main characteristics) of the car types that most usually transport the product. This approximation may give mixed results for those commodities, e.g. general cargo, that are moved by several significantly different car types. Good results are obtained for commodities using either a limited number of car types or dedicated cars: minerals, petrochemical products, refrigerated foods, automobiles, etc. As this second product group usually forms the main rail traffic, this approximation is satisfactory. In particular, for each product $p \in P$, let w_p be the quantity of product p that may be loaded into the corresponding vehicle p and vw_p the weight of an empty vehicle p.

Consider a link a representing either a regular link, a connector, a transfer, or even a yard. Flows v_a^p of products $p \in P$ travel on this link. An equivalent number of rail cars, n_a^p for each product $p \in P$, is then:

$$n_a^p = \frac{v_a^p}{w_p}$$

To estimate t_a, the total train flow on link a , one may then simply use an average number of cars per train **n** :

$$t_a = \sum_{p \in P} \frac{n_a^p}{n}$$

which is an alternative quite often chosen in tactical rail models (e.g. Petersen and Fullerton, 1975, Assad, 1980b, etc.).

This "average" approach is a very rough approximation of the processes that lead to the determination of the different train services, of their various loads, of their frequencies and corresponding train lengths. First, the resulting train flows are real valued and, generally, grossly under-estimate the actual traffic. When, on the other hand, the computations are done in integer arithmetic (e.g., Assad, 1980b), the results largely over-estimate the actual traffic. Besides, this approach does not take into account any information on the type of products transported or on the physical path used to move the freight, and the aggregation effect is even stronger when computing in real arithmetic. Consequently, the train composition and length usually are the result of trade-offs between economic and service criteria, product characteristics and physical constraints such as the maximum train length (imposed by the length of the sidings; in meters or number of cars) and the maximum train weight (due, mainly, to the infrastructure specifications). These trade-offs are normally included in tactical and operational models of rail operations.

In STAN, we searched for an approximation that is at the same time more aggregated than the tactical planning approaches and more precise than the simple "average" formula. The approximation is real valued and is computed locally, for each link, but it includes information on both product and link characteristics. The number of trains traveling on a particular link a is thus computed by the following formula:

$$t_a = \sum_{p \in P} \frac{v_a^p + (v_a^p / w_p) * vw_p}{\alpha_p * \beta_a}$$

where,

t_a : train flow on link a;

v_a^p : flow of product p on link a;

w_p : quantity of product p that may be loaded into the corresponding vehicle p;

vw_p : weight of an empty vehicle p;

α_p : weight of a train of product p;

β_a : link correcting factor.

This formula computes an *average train* for each product moving on the link and, then, combines them into the actual total link train flow, ajusting it for the characteristics of the particular link. The presence of both a product and a link transformation factor makes for increased flexibility and accuracy. In fact, our experiments, perfomed by using data from the Canadian National Railways and exposed in Chapter 7, clearly show that the "by product" approximation significantly outperforms the "average" approach, even when only the product adjusting factors (α_p) are calibrated and $\beta_a = 1$ for links. We have also found that this approximation gives an excelent fit in almost all cases. Slightly less accurate results are obtained only on links where small volumes of traffic are moved by a number of trains significantly larger than those normally corresponding to the intensity of the car and product flows. This, however, is an extreme condition, usually not present on the main lines and that is easily taken care of by using the link correcting factor. See Chapter 7 on the problem of the calibration of the individual α_p.

From a modeling point of view, it is interesting to note that different α coefficients may be used for one product by defining an adequate number of different commodities, all representing the same initial product, and by splitting the demand accordingly. This may be useful if one wants, for example, to discriminate among different origin-destination pairs. Practically however, this may rapidly and significantly increase the number of products, and thus the size of the problem, and caution should be exerted accordingly. As for the link correcting factor, it may be used to calibrate the train approximation formula to an "exact" fit for all links. This is an option that would, however, request an unacceptably long calibration effort in the case of most realistically sized networks. One would rather use the β_a factor to correct the estimation in extreme cases, as previously, or to "personalize" some parts of the network: lines of particularly low or high weight tolerance, subnetworks equipped with better traffic control and signalization equipment, etc.

The resulting estimation of the total link train flows are then used to account for the operating and energy consumption costs and to compute the line and yard delays generated by the traffic congestion and the limited facility capacities.

4.3.3 Delay estimation

Rail facilities, as those of other transportation modes, have physical capacities. There is an upper limit on the number of units (tons, cars, trains) that may pass through the facility during a specified time period. Well before the capacity of the facility is attained, however, this passing-through time, or *delay* , increases intolerably. This is caused by physical or operational bottlenecks and, when it occurs, it usually generates the redirection of the train traffic and, sometime, justifies the study of some infrastructure modifications. These two related notions of performance, *facility capacity* and its associated *passing-through delay* , are among the main considerations at all levels of rail transportation modeling.

Delay functions are most important because they allow to analytically describe the effect of the traffic intensity on the facility by the time necessary to pass through the facility. Figure 6 illustrates the general shape of such a delay function: beginning with a *free running time* , it increases slowly, almost linearly (the *uncongested* or *linear* part) at first, until the facility starts to be congested and the delay strongly increases with the traffic intensity (the *congested* component) toward an asymptotic infinity at the facility's capacity. This is a typical delay function derived from queueing theory models.

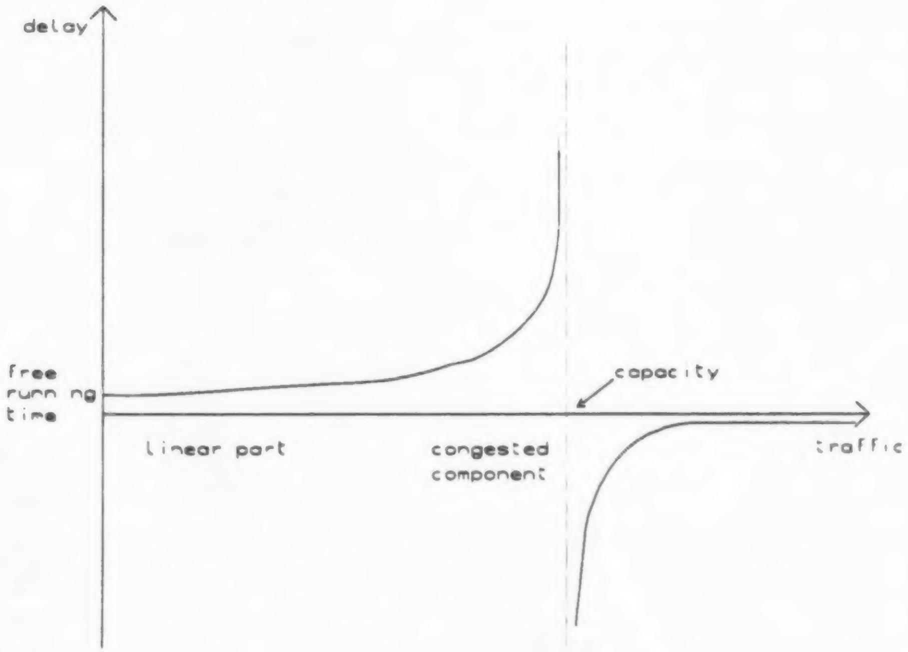


Figure 6 General Shape of a Delay Function

Consequently, in models for tactic and strategic planning, capacities may not be explicitly considered and are replaced by delay functions, provided that precautions are taken concerning their asymptotic behaviour. This is the approach of the present rail model and, in general, of STAN.

Delay functions are also important because they permit a better estimation of the total trip time, from the origin of the traffic to its destination. For instance, if modal choice is based on total trip time or, if specific ***commodity delay costs*** are defined accordingly, one may model different priority classes for the products and do the assignment following this criterion. And, with a reliable estimation of transportation time, vehicle depreciation costs may also be more easily computed.

Delay functions may be defined in yards, at transfers and over the lines of the network.

The various yard operations are usually grouped into five main categories: train reception, inspection and dismantlement, car classification and block formation, train make-up, waiting for train connection, train inspection and departure. Models are then fitted for each one of these main operations to estimate the yard's performance mesures. Various queueing models are usually used for this purpose (Crainic, 1987). For our model, we use a one-link representation of a classification yard and, therefore, we concentrate all these yard operations and delays into one function representing the *total yard put-through delay* . We then assume that the whole yard operation may be represented by an $M/M/1$ queue, whose mean waiting time represents the yard's average put-through delay. For a yard we then have:

$$Y_a = \frac{s_a * lp}{lp - s_a * \sum_{p\in P}(v_a^p/w_p)}$$

or, if one notes that, usually, $s_a = lp/CAP_a$,

$$Y_a = \frac{lp}{CAP_a - \sum_{p\in P}(v_a^p/w_p)}$$

where,

Y_a : Put-through yard delay for yard a;

v_a^p : Volume of traffic of product p transiting through yard a;

s_a : Average service ("classification" or put-through) time for yard a;

CAP_a : Capacity of yard a;

lp : Length of the planning period.

This formula has two major drawbacks:

i) The asymptotic behavior (toward infinity) of the yard delay when the total car flow through the yard approaches the yard capacity;

ii) The convexity of the objective function for the general model cannot be formally demonstrated when it includes this type of delay formula.

Empirically, the average service time and capacity may be quite difficult to evaluate precisely in most cases. For yards there is not, as there is for rail lines (e.g., Colson's formula), an accepted method that may be used to compute, under simplifying assumptions , a reasonably accurate estimate of their capacities; often, only historically observed yard put-through values are available.

Consequently, when the yard is not a bottleneck in a given network, nor is it forecast to became one in a future scenario, a constant (estimated from observed values) delay function may be used for Y_a. The capacity may then be introduced in that function as a limiting factor, by, for example, penalty methods (see Chapter 8). Otherwise, the yard put-through delay defined above should be taken as a starting point and a polynomial delay $\bar{Y}_a$ should be approximated using the procedure described below.

On the lines of a rail network, train meetings and overtakes can occur only at prespecified points in the network: stations or sidings. Delays are thus incured, and they increase with the traffic intensity and complexity.

A number of hypotheses on the nature of the train traffic were assumed for our model:

a) Passenger trains are not present on the network;
b) All freight trains have equal priority;
c) On a link, all trains moving in the same direction, travel at the same (and constant) speed;
d) Consequently, no more than two trains meet at the same time;
e) On each link, the traffic is globally balanced, meaning that the number of trains traveling in each direction is approximately the same.

These hypotheses were inspired by the context of the brazilian transportation system and by the needs of strategic planning and are appropriate for a large variety of applications. Given these hypotheses, no train overtakes may occur on the line of the network and the over-the-line delay for a train traveling on a double track line is equal to the *free running time* of the corresponding link. In this case, the line capacity may also be included in the delay function by using penalty methods (Chapter 8).

The average over-the-line delay for a train traveling on a single track link may be computed by different formulae according to the degree of detail that is sought after (Crainic, 1987). Given the strategic context of the model and our hypotheses, we employ a formula, first proposed by Petersen (1974), that makes use of the capacity of the line and the train flows in each direction:

$$D_a = \frac{FRT_a * CAP_a + (FRT_{\bar{a}} * t_{\bar{a}} - FRT_a * t_a)}{CAP_a - (t_{\bar{a}} + t_a)}$$

where the two links a and $\bar{a}$ represent the two directions of travel on a single track rail line and where,

D_a : delay on link a;
CAP_a : capacity of link a;
FRT_a : free running time on link a;
$FRT\bar{a}$: free running time on link $\bar{a}$;
t_a: train flow on link a;
$t_{\bar{a}}$: train flow on link $\bar{a}$.

We identified several problems related to the above formula. The first two

i) Asymptotic (toward infinity) behavior of the delay when the total train flow tends toward the link capacity (see Figure 6);

ii) The convexity of a general objective function that includes this delay formula cannot be formally demonstrated;

are common to this class of delay functions and were also identified for the yard delay formula. There is another one that is even more disturbing in the present context:

iii) When fully developed, by substituting the equation for the number of trains, the formula consists of too many terms, implying a more difficult user interface and much more time consuming algorithms, particularly the shortest path procedure.

We propose therefore a polynomial approximation of the delay function:

$$\bar{D}_a = FRT_a * (1 + \phi_1 * T_a + \phi_2 * (T_a/CAP'_a)^{\text{exp}})$$

where,

$\bar{D}_a$: estimate (approximation) of delay on link a;

T_a : total train flow on the link

$$T_a = t_a + t_{\bar{a}}$$

CAP'_a : value of the total train flow where the line begins to be considered congested;

and ϕ_1, ϕ_2, and exp are parameters to be calibrated.

The parameters could be calibrated by non-linear regression methods and we examine this option in Chapter 7. This approach is not, however, entirely satisfactory for real applications, since too large a number of different polynomials would be generated (in fact, a different polynomial for each link of the network!). Also, the degree of the polynomials generated by non-linear regression methods are, generally, quite high, resulting in possible numerical problems and longer solution times for the algorithm.

A heuristic procedure may rather be used to calibrate the parameters of the polynomial approximation. The procedure estimates values for the three parameters, exp, ϕ_1 and ϕ_2, of the polynomial delay $\bar{D}_a$, such that it fits closely to the delay D_a computed by a given formula (e.g. Peterson's link delay (1974), yard classification delay, or other similar function).

For link **a** , the general procedure is:

1) Generate a set of points (T_a, D_a) = (total train traffic, delay), where $T_a = t_a + t_{\bar{a}}$, for various combinations of train flows t_a et $t_{\bar{a}}$. Fix the value of CAP'_a at the level of the total train flow where the link "begins" to be considered congested.
2) Plot the points obtained in step 1, "observe" the curvature of the scatter diagram and fix exp accordingly.
3) Estimate ϕ_2 by fitting $D_a = FRT_a * (1 + \phi_2 * (T_a/CAP'_a)^{\text{exp}})$ for high values of T_a, i.e. $T_a \geq CAP'_a$.
4) Evaluate ϕ_1 to adjust $D_a = FRT_a * (1 + \phi_1 * T^*_a + \phi_2 * (T^*_a/CAP'_a)^{\text{exp}})$ for a point in the linear part of the scatter diagram, e.g. $T^*_a \in [0,1 * CAP_a, 0,2 * CAP_a]$, and the value of ϕ_2 calculated at step 3.
5) Repeat steps 2 to 4 for "intelligent" choices of exp and choose the best fit polynomial.

We have tested this procedure with excelent results (see Chapter 7), even when compared to estimating the parameters directly with nonlinear regression methods. The same chapter also presents some practical suggestions on how best to use the procedure, especially for the links of a realistically large network.

4.4 EMPTY FLOW ESTIMATION

An important aspect of any freight transportation system is the presence of a significant, both in terms of volume and costs, traffic of empty vehicles. This traffic is usually generated by the imbalances in the transportation demand, the fact that specialized vehicles are used for certain products and the need to relocate the vehicles in view of the transportation requirements of the following periods.

The impact of empty vehicle movements on both the operations and planning of transportation systems, which is significant for all transportation modes, is even more important for the rail mode, since rail vehicles are captive to the infrastructure and thus, the empty car movements occur on the same rail lines as the loaded ones. Often, empty and loaded cars are mixed together to form blocks and trains. Consequently, the train flows and the over-the-line delays are significantly increased and, therefore, the proper modeling of strategic or tactical rail planning has to include some form of representation of empty car movements.

Since a strategic model of the freight rail transportation system, such as STAN, considers only product flows and estimates the train congestion at the link level, one must include a mechanism to allocate to each link a certain amount of empty car movements to be considered, while computing the train flow and the corresponding over-the-line delay.

The conditions that we impose on such a mechanism specify that it should use only the information currently available in the model (i.e., the loaded transportation demand and current traffic pattern) and that the resulting empty car allocation should change with the variations in the routing of the loaded traffic during the solution of the multimodal multicommodity assignment problem.

A relatively important number of models have already been proposed for the empty vehicle transportation problem, for all modes and at various planning levels. Dejax and Crainic (1987) present a thorough review and analysis of these models. For our part, we decided to use a procedure based on a gravity model similar to the the one successfully used by Coutu (1978) for the forecasting of the empty rail car traffic for the Canadian National Railways. We first present the general approach and the details of the gravity model, followed by the statement of the calibration procedures for the parameters of the model. We conclude with the presentation of two possible utilization modes for the empty flow estimation procedure in strategic analyses to be performed by STAN in practice.

4.4.1 The procedure

The link flows in our model are the result of the assignment procedure in STAN. Consequently, the most effective and convenient approach to treat the empty car flow estimation problem in concordance with the general principles of STAN is to define the empty cars as a separate *product* .

The product *empty car* will then be transported, to satisfy a certain origin-destination demand, concurently with the other products, thus contributing to the general network congestion, delays and costs. The general procedure is:

1) Define the *empty car* product and its transportation characteristics;
2) Estimate the origin-destination demand matrix for this product;
3) Include this demand in the general model and treat it as any other product.

To define a product, one has to name it and to specify, for each mode, the characteristics of the vehicle that is used to transport it, as described in Section 4.2.

Theoretically, one could define an *empty car* commodity for each one of the products and vehicles present in a particular application. This would generate, however, a large number of additional products that would very rapidly became prohibitive in terms of memory and computing time requirements. It should be noted that we are not interested in the exact reproduction of the actual car traffic in the network and, also, that the vehicles associated with products in STAN are already "average" or "typical" vehicles, one per product. Consequently, *one* product should be enough for the estimation of the link empty car flow and the rest of this section is based on this hypothesis. If it became necessary to consider several *empty car* products, the same procedure applies for each one of these.

Our *empty car* product is identified by the ω character. Its associated vehicle for the rail mode (the product ω is not transported on any mode that is not rail based) is a combination of the "loaded" vehicles already defined for the transportation, by the rail modes, of the products of a specific application. Its loading factor, w_ω , equals the weighted average of the w_p values that correspond to the loading factors of these vehicles. As for its empty weight, vw_ω , it is fixed at zero to avoid a double counting in the train estimation formula. The various unit costs usually associated with products and their *typical* vehicles, may be, for the *empty car* product, either exogenously defined, or derived, by using weighted averages, from the corresponding attributes of the "loaded" vehicles.

To estimate the origin-destination demand of empty movements, one could use a variety of heuristic and exact methods (Dejax and Crainic, 1987). We adopt the use of a *gravity* model (Coutu, 1978; Nguyen, 1984) that assumes that the empty flow between a given origin-destination pair is proportional to the supply at the origin and the demand at the destination; the proportionality factor usually depends on parameters such as distance and the cost of travel between the two centroids. The formulation also assumes that the allocation of empty car movements among the different origin-destination pairs is stable from one planning period to another. The model is:

$$\begin{array}{lrl}
\textit{Determine} & g^{\tau}_{od} = \gamma_0 * \theta_d * r_k & \forall o \in O, \forall d \in D \\
\textit{Subject to} & \sum_{o \in O} g^{\tau}_{od} = D^{\tau}_d & \forall d \in D \\
 & \sum_{d \in D} g^{\tau}_{od} = S^{\tau}_o & \forall o \in O \\
 & g^{\tau}_{od} \geq 0 & \forall o \in O, d \in D
\end{array}$$

where,

g^{τ}_{od} : The origin-destination demand to be determined, i.e. the number of empty cars to be moved from origin $o \in O$ to destination $d \in D$ in period τ;

γ_0, θ_d : Balancing coefficients;

r_k : Balancing coefficient for distance (cost) class k;

S^{τ}_o : Supply of empty cars at origin $o \in O$, in period τ;

D^{τ}_d : Demand of empty cars at destination $d \in D$, in period τ.

The model determines the demand for each origin-destination pair such that the total supply is shipped out of surplus centroids while each deficit centroid receives its total demand. The allocation is done according to the class k to which belongs the origin-destination pair following prespecified criteria (cost, distance, flows, etc.). Thus, the corresponding r_k factors represent weights that indicate, with respect to a *base year*, the relative importance of each class and, therefore, of the empty flow to be allocated to each origin-destination pair in that class. The allocation may also be ajusted for known, or forecasted, variations of supplies and demands at centroids through the *balancing* coefficients γ_0 and θ_d. These are obtained from a balancing procedure with predicted S^{τ}_o and D^{τ}_d.

We discuss in the following sections the calibration of the distance balancing coefficients and the estimation of the supplies and demands at centroids for the planning period under consideration. Assuming that these quantities are known, however, the *prediction model* may be solved by a two-dimensional balancing procedure (see Nguyen (1984) for details on this type of procedure):

STEP 0. For all origin-destination pairs (o,d), $o \in O$, $d \in D$, let $g^{\tau}_{od} = r_k$ if (o,d) belongs to distance class k, 0 otherwise;

STEP 1. For all $o \in O$, let $g^{\tau}_{od} = \dfrac{g^{\tau}_{od} * S^{\tau}_{o}}{\sum_{d \in D} g^{\tau}_{od}}$

STEP 2. For all $d \in D$, let $g^{\tau}_{od} = \dfrac{g^{\tau}_{od} * D^{\tau}_{d}}{\sum_{o \in O} g^{\tau}_{od}}$

STEP 3. If $[\dfrac{S^{\tau}_{o}}{\sum_{d \in D} g^{\tau}_{od}} \approx 1]$ and $[\dfrac{D^{\tau}_{d}}{\sum_{o \in O} g^{\tau}_{od}} \approx 1]$
STOP, else go to STEP 1.

The result of this procedure is the empty car demand matrix g^{τ}_{od} for the planning period τ, and the balancing coefficients.

4.4.2 Supply and demand estimation

The supplies and demands of empty cars at the centroids of the network are estimated as the difference between the number of arriving and departing loaded cars at each centroid. According to the existing information, there are two slightly different computation methods based on this principle:

CASE 1. Product volumes are known, possibly following a previous assignment, on the links of the network. Then, if A^{+}_{j} represents the set of connectors going out of centroid j, while A^{-}_{j} stands for the set of connectors going into centroid j, the supplies at origins o, $o \in O$, are computed as

$$S_o^\tau = \max\,\{0, \sum_{A_o^-} \sum_{p \in P} (v_a^p / w_p) - \sum_{A_o^+} \sum_{p \in P} (v_a^p / w_p)\}$$

while the demands at destinations d, dD, are equal to

$$D_d^\tau = \max\,\{0, \sum_{A_o^+} \sum_{p \in P} (v_a^p / w_p) - \sum_{A_o^-} \sum_{p \in P} (v_a^p / w_p)\}$$

CASE 2. No information is available on product flows, only the total product supplies and demands at centroids are known. Then, the empty car supplies and demands are computed by the following formulae:

$$S_o^\tau = \max\,\{0, \sum_{p \in P} \sum_{d \in D} g_{od}^p / w_p - \sum_{p \in P} \sum_{o \in O} g_{od}^p / w_p\}$$

$$D_d^\tau = \max\,\{0, \sum_{p \in P} \sum_{o \in O} g_{od}^p / w_p - \sum_{p \in P} \sum_{d \in D} g_{od}^p / w_p\}$$

4.4.3 Model calibration

Various *costs* , or *impedances*, may be used to determine the r_k coefficients for the allocation of the empty flows to the different origin-destination pairs: travel times, transportation costs, previous flows, etc. Following the method of Evans and Kirby (1974), the domain of the measure of impedance is divided in several intervals, or classes, and if the impedance of an origin-destination pair falls in interval k then the pair belongs to class k and its cost is the coefficient r_k. The lengths of the intervals are application specific but should not be too small to attenuate the impact of variations between the **base year** and the planning period under study.

Best results were obtained by Coutu (1978) when the mesure of impedance was the inverse of the observed empty flows between the origin-destination pairs. Then, between an origin-destination pair $(o,d), o \in O, d \in D$, a flow g_{od}^{b} of empty cars was observed in the *base year* **b**, and the pair is said to belong to class C_k if

$$\frac{1}{g_{od}^{k-1}} < \frac{1}{g_{od}^{b}} \leq \frac{1}{g_{od}^{k}}$$

The calibration of the r_k cost coefficients is performed by assuming that the distribution of the empty car demand in the *base year* verifies the following model:

$$\begin{array}{lrl}
\textit{Determine} & g_{od}^{b} = \bar{\gamma}_0 * \bar{\theta}_d * r_k * \delta_{od}^{k} & \forall o \in O, \forall d \in D, \forall k \\
\textit{Subject to} & \sum_{o \in O} g_{od}^{b} = D_d^{b} & \forall d \in D \\
 & \sum_{d \in D} g_{od}^{b} = S_o^{b} & \forall o \in O \\
 & \sum_{C_k} g_{od}^{b} = R_k^{b} & \forall k \\
 & g_{od}^{b} \geq 0 & \forall o \in O, d \in D
\end{array}$$

where,

δ_{od}^{k} : 1 if $(o,d) \in C_k$ and 0 otherwise;

R_k^{b} : Total empty car traffic observed for class k;

$$R_k^{b} = \sum_{C_k} g_{od}^{b} \qquad \forall k$$

S_o^{b} : Total observed empty car traffic out of origin o

$$S_o^{b} = \sum_{d \in D} g_{od}^{b} \qquad \forall o \in O$$

D_d^{b} : Total observed empty car traffic in destination d;

$$D_d^{b} = \sum_{o \in O} g_{od}^{b} \qquad \forall d \in D$$

A three-dimentional balancing procedure is then used to determine the r_k coefficients and thus calibrate the gravity model. Note that the estimation of g_{od}^{b} is a by-product of the calibration procedure that is not needed afterwards. The procedure:

STEP 0. Let $r_k = 1$ for all classes k;
For all origin-destination pairs (o,d), $o \in O$, $d \in D$, let $g_{od}^{b} = \delta_{od}^{k} * r_k$

STEP 1. For all $o \in O$, let $g_{od}^{b} = \dfrac{g_{od}^{b} * S_o^{b}}{\sum_{d \in D} g_{od}^{b}}$

STEP 2. For all $d \in D$, let $g^b_{od} = \dfrac{g^b_{od} * D^b_d}{\sum_{o \in O} g^b_{od}}$

STEP 3. For all classes k, let $r_k = \dfrac{r_k * R^b_k}{\sum_{C_k} g^b_{od}}$ and $g^b_{od} = \dfrac{g^b_{od} * R^b_k}{\sum_{C_k} g^b_{od}}$

STEP 4. If $[\dfrac{S^b_o}{\sum_{d \in D} g^b_{od}} \approx 1]$ and $[\dfrac{D^b_d}{\sum_{o \in O} g^b_{od}} \approx 1]$ and $[\dfrac{R^b_k}{\sum_{C_k} g^b_{od}} \approx 1]$
STOP, else go to STEP 1.

The resulting r_k coefficients may then be used, with the estimations of the total supplies and demands, to forecast the origin-destination demands for empty car travel.

4.4.4 Utilization modes

STAN is usually used to analyse future, forecasted or hypothetical, situations and to compare the results of these scenarios to some given base year data. It is thus of interest to examine how to use the empty flow estimation procedure in these cases.

One should first note that data from a *base year* is always required to calibrate the r_k cost coefficients and that following that operation the *balancing* coefficients may be used to ajust the relative weights of the centroids to the forecasted environment. Secondly, it should also be noted that one may use STAN in either of two modes: to perform a *multiperiod* analysis or a *one period* study.

For a *multiperiod* analysis, an assignment is performed for each period and the demand matrices, and eventually the network, are ajusted before each new period. Then, the CASE 1 of Section 4.4.2 applies and the supplies and demands of empty cars for a new period are estimated from the product flows determined at the previous period. Additionally, the empty car demand matrix of the previous period is used as the *base year* data. When, on the other hand, one performs a *one period* study, this information is not available and the estimation has to be done as described in the second case of Section 4.4.2. It is obvious that, in this situation, the overall empty flow estimation would not be very accurate: the r_k cost coefficients are calibrated using data that may be quite different from that of the current scenario, and the total empty car supply and demand for the period are estimated using the transportation demand for loaded movements for the same period. However, this is a far better solution than that adopted by most strategic planning models that choose to ignore the empty flow problem altogether.

We want finally to emphasize that, when information is available on certain empty car movements, such as the knowledge that the cars moving a certain product between a given origin-destination pair are always going back empty to the origin, this information should be included in the empty flow estimation model by fixing the corresponding demand. That would result in a more accurate prediction and better overall results.

4.5 BIBLIOGRAPHY

Assad, A.A. (1980a), *Models for Rail Transportation*, **Transportation Research** 14A, 205-220.

Assad, A.A. (1980b), *Models of Rail Networks: Toward a Routing/Make-up Model*, **Transportation Research** 14B, 101-114.

Bronzini, M.S. (1980), *Evolution of a Multimodal Freight Transportation Network Model*, in *Proceedings of the 21st Annual Meeting*, **Transportation Research Forum**, 475-485.

Bronzini, M.S. and Sherman, D. (1983), *The Rail Carrier Route Choice Model*, **Transportation Research** 17A (6) 463-469.

CNR (1983), *Rapport annuel 1983*, Canadian National Railways.

Crainic, T.G. (1982), *Un modéle de planification tactique pour le transport ferroviaire des marchandises*, Ph.D. Thesis, publication #247, Centre de recherche sur les transports, Université de Montréal.

Crainic, T.G. (1987a), *Operations Research Models of Intercity Freight Transportation: The Current State and Future Research Issues,* **The Logistics and Transportation Review** 23(2), 189-206.

Crainic, T.G. (1987b), *Rail Tactical Planning: Issues, Models and Tools*, **Freight Transport Planning and Logistics**, Springer Verlag, L, Bianco and A. La Bella (editors), 463-509.

Crainic, T.G., Ferland, J.A. and Rousseau, J.-M. (1984), *A Tactical Planning Model for Rail Freight Transportation*, **Transportation Science** 18 (2), 165-184.

Dejax, P. and Crainic, T.G. (1987), *A Review of Empty Flows and Fleet Management Models in Freight Transportation*, **Transportation Science** 21(4), 227-247.

Friesz, T.L., Gottfried, J.A. and Morlok, E.K. (1986), *A Sequential Shipper-Carrier Network Model for Predicting Freight Flows*, **Transportation Science** 20 (2), 80-91.

Jones, P.S. and Sharp, G.P. (1979), *Multi-Mode Intercity Freight Transportation Planning for Underdeveloped Regions*, in *Proceedings of the 20st Annual Meeting*, **Transportation Research Forum** .

Kornhauser, A.L., Hornung, M., Hazony, Y. and Lutin, J. (1979), *The Princeton Railroad Network Model: Application of Computer Graphics in the Analysis of a Changing Industry,* presented at the 1979 Harvard Graphics Conference.

Kresge, D.T. and Roberts, P.O. (1971), *System Analysis and Simulation Models*, in *Techniques of Transport Planning*, John R. Meyers ed., vol. 2, Brookings Institute, Washington, D.C.

Lansdowne, Z.F. (1981), *Rail Freight Traffic Assignment*, **Transportation Research** 15A, 183-190.

McGinnis, L.F., Sharp, G.P. and David Yu, H.C. (1981), *Procedures for Multi-State, Multi-Mode Analysis: Vol. IV, Transportation Modeling and Analysis*, U.S. D.O.T. report no. DOT-OST-80050-17/V.IV.

Nguyen, S. (1984), *Estimating Origin-Destination Matrices from Observed Flows*, in *Transportation Planning Models*, M. Florian (Editor), Elsevier Science Publishers B.V. (North Holland), 363-380.

Petersen, E.R. (1974), *Over-the-road Transit Time for a Single Track Railway*, **Transportation Science** 8, 65-76.

Petersen, E.R. (1977a), *Railyard Modelling: Part I. Prediction of Putt-through Time*, **Transportation Science** 11 (1), 37-49.

Petersen, E.R. (1977b), *Railyard Modelling: Part II. The Effect of Yard Facilities on Congestion*, **Transportation Science** 11 (1), 50-59.

Petersen, E.R. and Fullerton, H.V., editors (1975), *The Railcar Network Model*, Report No. 75-11, CIGGT, Queen's University, Kingston, Ontario.

Roy, M.-C. (1986), *Modélisation stratégique du transport des marchandises au Brésil: le mode ferroviaire et l'analyse de matrices de demande,* M. Sc. thesis, publication #485, Centre de recherche sur les transports, Université de Montréal.

Sharp, G.P. (1979), *A Multi-Commodity, Intermodal Transportation Model*, in *Proceedings of the 20st annual meeting*, **Transportation Research Forum**.

CHAPTER FIVE

MODELING OF OTHER TRANSPORTATION MODES

A multi-modal transportation network, of a country or a nation, usually consists of several modes besides rail, even though modeling the rail mode is a more challenging task. All the present transportation modes, such as road, navigation, etc., have to be represented in a complete network model.

In most cases, the road mode plays a major role, as trucks transport a large proportion of freight, measured either in tonnage or value. The navigation modes (internal, coastal and maritime) also have to be considered, as well as various other means of carrying freight, such as pipelines, air, etc. Finally, intermodal transfer facilities and ports form an important component of the transportation network and must be present in the network model as special modes.

This chapter presents the modeling principles used in STAN for the representation of the *road, port* and *navigation* modes. A number of important issues, that have to be considered if other modes are to be included in the model, are discussed as well.

5.1 THE ROAD MODE

Even though the road mode is one of the most important in terms of the quantities of goods shipped, it is relatively straightforward to represent in the STAN system by using the standard components of the transportation network, which are the nodes, centroids, directional modal links and transfers.

The road infrastructure is thus represented by directional modal links, where nodes represent cities or intersections of roads. Additional nodes may be used, where necessary, to increase the topological similarity with the actual road system or to add connection links from centroids to the road network. For strategic planning purposes, most of the links of the road mode are not subject to congestion, although volume/delay curves, which model congestion, may be specified for links which are near to major urban agglomerations, where road vehicles may exprience delays due to heavy traffic.

As the road mode may be used to transport freight to or out of a rail terminal or a port, where the goods are transshipped between the raod and a different mode, the nodes that have adjacent links of the road and of another mode should have transfers specified with the appropriate attributes.

The cost associated with each link and transfer in the STAN system may have time delays, operating costs and energy consumption characteristics. These costs are related to the characteristics of the links (geometry, condition of road) and of the vehicles that are associated with the road mode. In effect, with each product defined in the STAN data base that is transported on the road mode, there is an "average" or "typical" vehicle associated. The characteristics of this vehicle determine the operating costs and the energy consumption rate. For each link of the network, one can define a function class, which gives the functions of travel time, operating cost and energy consumption for the various typical vehicles. Thus, the links of the network may be subdivided into categories and a function class may be defined for each category. The data necessary for evaluating these functions are usually obtained from surveys of the operating costs and fuel consumption of trucks over different types of links in the road network. The conditions of the road links are also usually available from surveys carried out periodically by the concerned governmental agencies (see Chapter 3, Part IV).

Finally, the time delay, costs and fuel consumption of the transshipment movements depend on the mode to or from which the freight is transfered and on the characteristics of the handleing equipement available at the transfer points. Some of these aspects are discussed in the section of this chapter that deals with the modeling of ports.

In summary, the modeling of the road mode and its associated network is rather straightforward in STAN. On the other hand, it is not trivial to collect and analyse the data required to properly classify the links of the network and to define the typical vehicle for each product transported on the road mode, chapter 3 of part IV, further examines these issues.

5.2 MODELING PORTS

Ports represent a major element of a national or regional multi-modal transportation system that has access to a sea or an ocean, or that comprises a significant river or lake navigation system. Import and export flows usually use the ports, as well as domestic flows that use the coastal navigation system. Port facilities, capacities and operations thus impact upon the performance of the entire network and a strategic model must capture these impacts.

We recognize this importance by including in STAN the possibility of a *port mode* to represent port infrastructure and operations. The challenge here is to abstract the complexity of port operations within a strategic planning model and the means provided in STAN to represent the transportation network. The presentation of port modeling is the aim of this section.

5.2.1 Port operations

Ports are complex instalations (see, for example, Figure XX, Chapter 4, Part IV), where a wide variety of transport, transshipment and storage activities are ongoing continuously. The precise nature of these operations, as well as their cost and time characteristics, are dependent on the type of infrastructure and equipment used for the various classes of commodity handled. The commodity types most frequently encountered include general cargo, grain, coal, ore, rolling cargo, liquid commodities, chemical products, containers, among others. These commodity classes may use either specialized and dedicated or common facilities. Storage may occur either in specialized or general purpose warehouses, or in the land vehicles used to carry the freight into ports.

Ports are multi-terminal facilities dedicated to the transshipment of commodities between maritime and land transportation modes. The typical port analysis is dynamic in nature. The aims are, either to maximize the port throughput, its revenues and profits, or to use its existing capacity at minimum cost. A more general system objective is to attain a "minimum" cost for the transportation system by simultaneously optimizing its port and land components (Wilness and Frankel, ??). In all cases, the cost and delay relationships between the freight volumes moved through the port installations and the characteristics of the available resources, offer model components and parameters vital for any representation of port operations.

Our strategic approach considers the ports as part of the entire transportation system. The representation is necessarily static and aims to capture the impact of a port's operations on the performance of the whole network, rather than to obtain a perfect reproduction of activities in a particular facility. The cost and delay aspects are, nevertheless, explicitly considered. Small or less important port installations may, however, be modeled as nodes with transfers, which is adequate to represent the associated intermodal transshipments.

When considering specific port operations one has to distinguish between *import* and *export* flows. Freight destined for export usually arrives into port in vehicles of a land mode (rail cars or trucks) or of a navigation mode (barges or coastal ships). After a certain waiting period, the vehicles are moved toward the unloading facilities. These installations may be mode specific or capable to service several modes, such as the grain unloading facilities that accept both rail cars and trucks. Freight is then transported either toward the pier and the ship or toward some storing facility to await the availability of loading equipment or the arrival of the ship. At that moment, the freight is taken out of warehousing and send to the pier for loading into the ship that transports it toward its destination, which may be within the country or a foreign port.

Freight that is imported is handled in a way that is symmetric to the export flows. Freight arrives by ship and is unloaded, either directly into land or navigation vehicles, or into warehouses. The equipment used to load the vehicles may again be specific for the transportation mode or be able to accomodate vehicles from several modes. It is also quite usual that the same equipment (cranes, grain elevators, etc.) be employed both to load and to unload a vehicle, truck, car, ship, etc., which gives rise to congestion and delay phenomena. Various additional delays may occur during this process: in ship before unloading, in vehicle before leaving port, etc. Once loaded into vehicles, the freight leaves for its inland destination.

Figure 1 illustrates these typical port operations. It displays the freight flows though a port terminal in a time-space diagram. Links 1 to 5 and 18 to 22 represent freight unloading from, loading into and stored in land vehicles. Links 10 to 13 represent the same freight movements for ships, while links 7 and 15 represent warehousing activities.

This representation, and the subsequent model, address the operations in one terminal only and tend to overlook interterminal movements. Such movements may occur either during the warehousing period or in conjunction with the unloading/loading activities, and are caused by the limited availability of warehouse capacity or of specialized equipment. Such operations, however, can not easily be considered at a strategic planning level, since, in the long term, it can be safely assumed that their effects on the performance of the port system are not sufficiently significant.

5.2.2 The network

The elements used to model the port network are the same as those of the general STAN network: modal links, regular nodes and centroids. The time dimension displayed in Figure 1 is represented by delay functions on the links of the network. Each link also has a cost function. Both delay and cost functions may be product specific.

Figure 2 illustrates the first version of the port network. It is a rather straightforward transposition of the movements defined in Figure 1, but without the time factor. Note that, for the sake of clarity, only three modes have been displayed: rail, road and navigation. Nodes 11 , 12 and 13 thus represent the connection between the port network and the land network for these three modes. Figure 2 displays movements for both the import and the export of freight flows. It also illustrates how, through node 3 and its incident links with appropriately defined delay functions, this model handles interperiod storage in a static framework.

This representation is still too detailed, when compared with the level of detail of a STAN multi-modal network. A number of simplifications have to be performed, and Figure 3 illustrates this process. The main simplifications concern the links that represent the movements of land vehicles inside the port and the in-vehicle storage of freight. The latter simplification may be easily justified by the relative unimportance of in-vehicle storage as compared with warehouse storage.

The aggregation of the links that represent movements of land vehicles was chosen based on consistency arguments. Land modes usually have associated vehicles. Yet, the port mode does not have any. Furthermore, the precise definition of modes may vary with the specific scenario. This implies that the port representation varies accordingly. Thus, it was decided that all modal movements should be considered

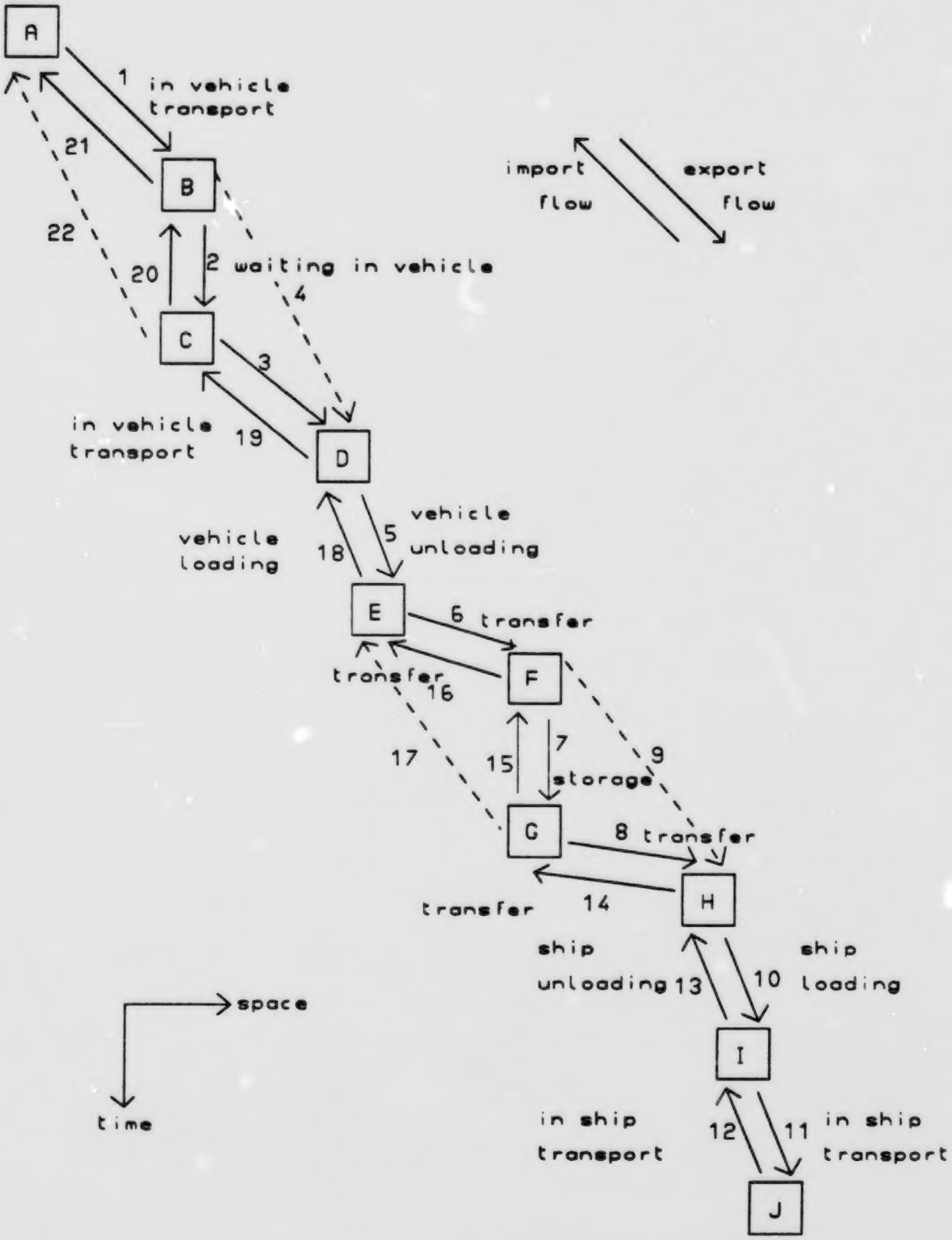

A
1 in vehicle transport
21
B
22
20
2 waiting in vehicle
4
C
3
in vehicle transport
19
D
vehicle loading
18
5 vehicle unloading
E
6 transfer
transfer
16
F
17
15
7 storage
9
G
8 transfer
transfer
14
H
ship unloading
13
10 ship loading
I
in ship transport
12
11 in ship transport
J
import flow
export flow
space
time

Figure 1 Time-Space Diagram of Freight Movements Through Ports

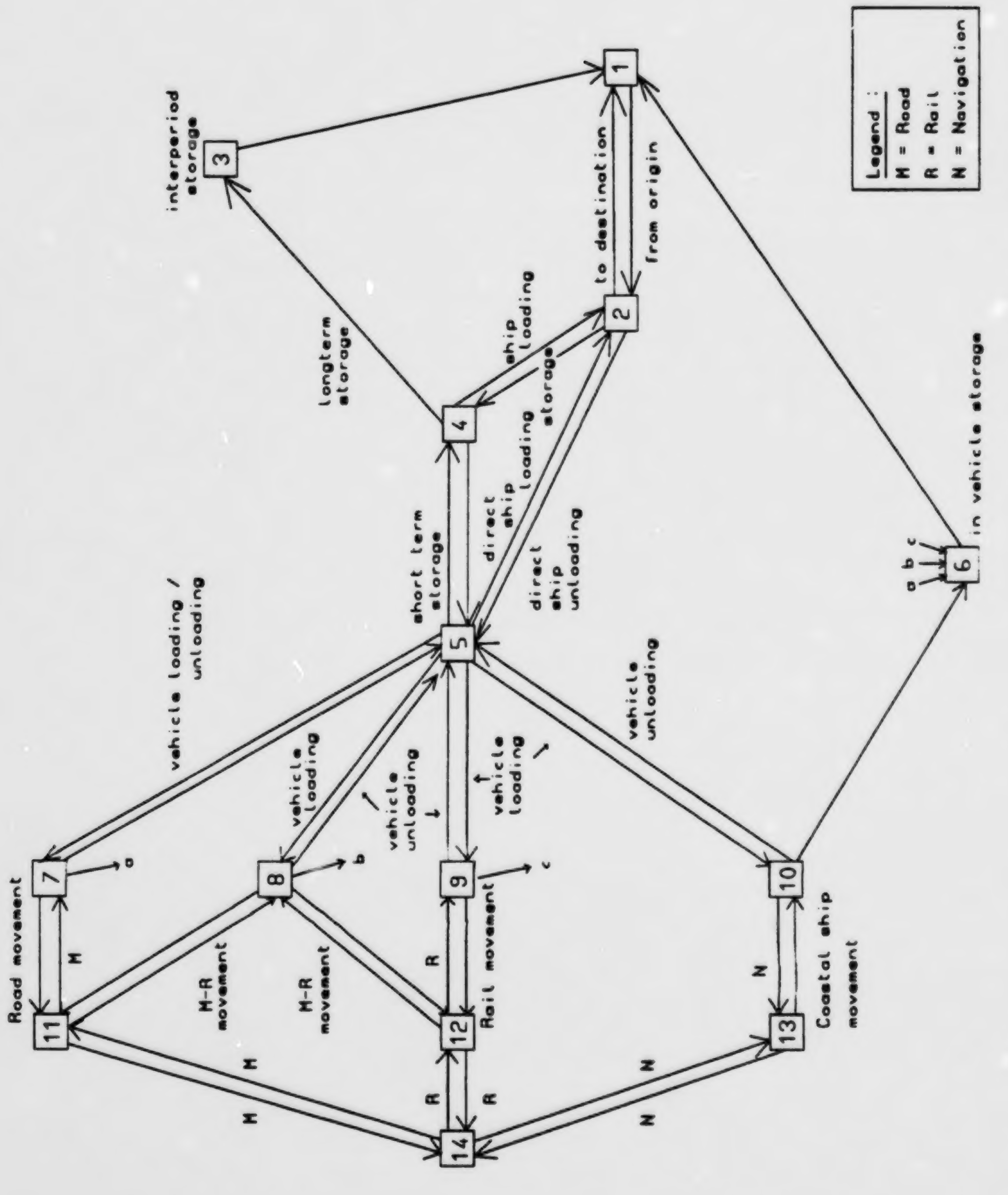

interperiod storage
3
1
to destination
from origin
2
ship loading
storage
longterm storage
4
direct ship loading
short term storage
direct ship unloading
5
6
a b c
in vehicle storage
vehicle loading / unloading
vehicle loading
vehicle unloading
vehicle loading
vehicle unloading
Road movement
7
a
8
b
9
c
10
M
M-R movement
M-R movement
R
Rail movement
N
Coastal ship movement
11
12
13
14
M
M
R
R
N
N
Legend :
M = Road
R = Rail
N = Navigation

Figure 2 General Network Representation of Port Activities

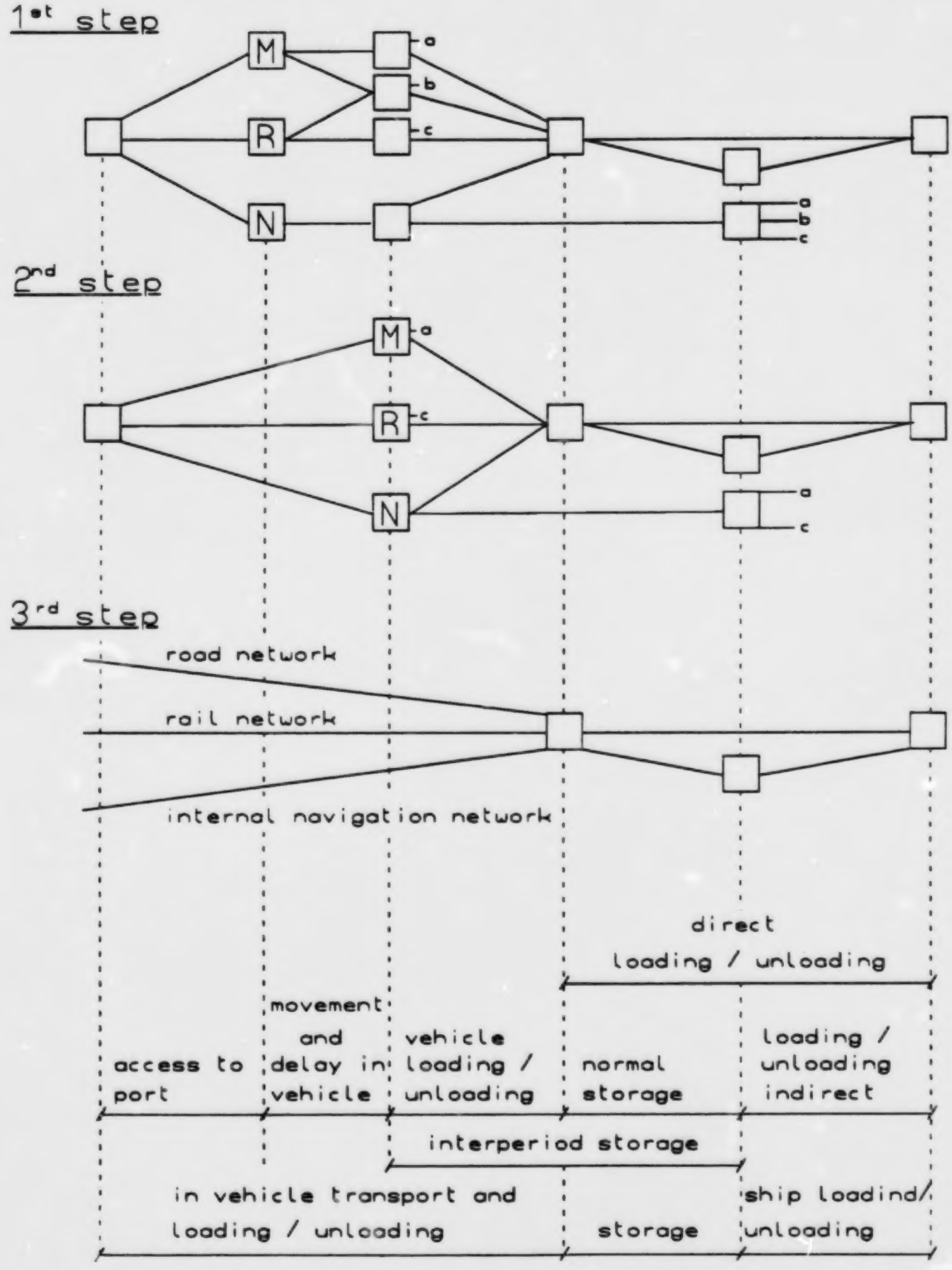

1st step
M
R
N
a
b
c
a
b
c
2nd step
M
a
R
c
N
a
c
3rd step
road network
rail network
internal navigation network
direct loading / unloading
access to port
movement and delay in vehicle
vehicle loading / unloading
normal storage
loading / unloading indirect
interperiod storage
in vehicle transport and loading / unloading
storage
ship loadind/ unloading

Figure 3 Evolution of the Port Network Representation

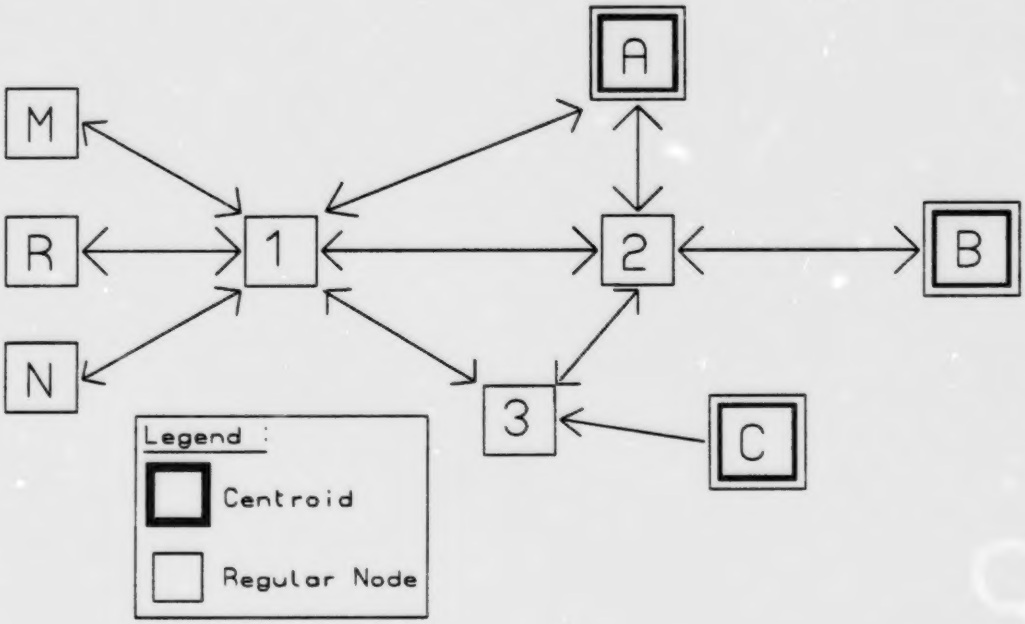


Figure 4 Final Model of a Port Network

outside the port, on the respective modal subnetworks. The delays associated with these movements may then be modeled by delay functions defined on the transfers between the port subnetwork and the links that connect it to the various modal subnetworks.

Figure 4 illustrates the final representation of a port network. Nodes 1, 2 and 3 are regular nodes, while nodes A, B and C are centroids. Nodes M, R and N are dummy nodes that represent the connection of the port network with the other modes: Road (Motor carrier), Rail and Navigation. Links (1,2) handle direct movements between ships and land vehicles, while links (1,3) and (3,2) are used to model storage activities. Links (1,A), (2,A), (2,B) and (C to 3) are connectors linking the centroids to the port network. Centroid A concentrates the offer and demand in the port, while centroid B represents an origin/destination outside the country. Centroid C is used, in a multiperiod analysis, as an origin for the freight stored in port at the end of a period, and shiped, during the following period, to a destination represented by either centroid A or centroid B. This feature is further discussed in Section 5.2.4.

The port network is connected to both the land transportation network and to other ports through navigation, coastal or ocean, links. We further investigate these links and subnetworks in Sections 5.2.3 and 5.3.

We examine now some possible flow movements through this simplified port network and their relations to actual port movements. This will help to determine the link and transfer delay functions that have to be specified.

5.2.3 Flows, transfers and delays

We first enumerate some of the movements that are feasible by using the port network model just described.

1) Export flows with destination in the port.
   - (1 - 2 - A): direct ship loading, without any storage in the port;
   - (1 - 3 - 2 - A): indirect ship loading, the goods are stored in the warehousing facilities of the port;
   - (1 - A): interperiod storage in port; the flows on link (1,A) represent the volume of traffic sent to the port in excess of its capacity;
   - (C - 3 - A): ship loading of the freight stored in port in previous periods.

2) Export flows with destination outside the port (here, centroid B represents either a foreign port, or a port connected to the internal/coastal navigation system).
   - (1 - 2 - B): direct ship loading;
   - (1 - 3 - 2 - B): indirect ship loading;
   - (C - 3 - B): ship loading of the interperiod stored freight.
3) Import flows that originate in the port.
   - (A - 2 - 1): direct ship unloading;
   - (A - 2 - 3 - 1): indirect ship unloading.
4) Import flows that originate outside the port.
   - (B - 2 - 1): direct ship unloading;
   - (B - 2 - 3 - 1): indirect ship unloading.

All possible movements on the port network illustrated in Figure 4 have not been enumerated. Those that have not been mentioned do not correspond to real activities and must be prohibited. Some movevements, such as indirect ship loading, may occur only when some specific conditions, e.g. insufficient ship capacity, occur. Transfers and delay functions, on both links and transfers, may advantageously be used to respond to these modeling requirements. Movement prohibition, in particular, may be inforced either by not defining the transfer between some specific modes (e.g. port and connector) at a node, or by specifying a theoretically infinite transfer delay, if the transfer is between two links of the same mode. To illustrate these concepts, one may examine, for example, the transfers at nodes 1 (see Figure 5) and 3 (Figure 6) of the port network.

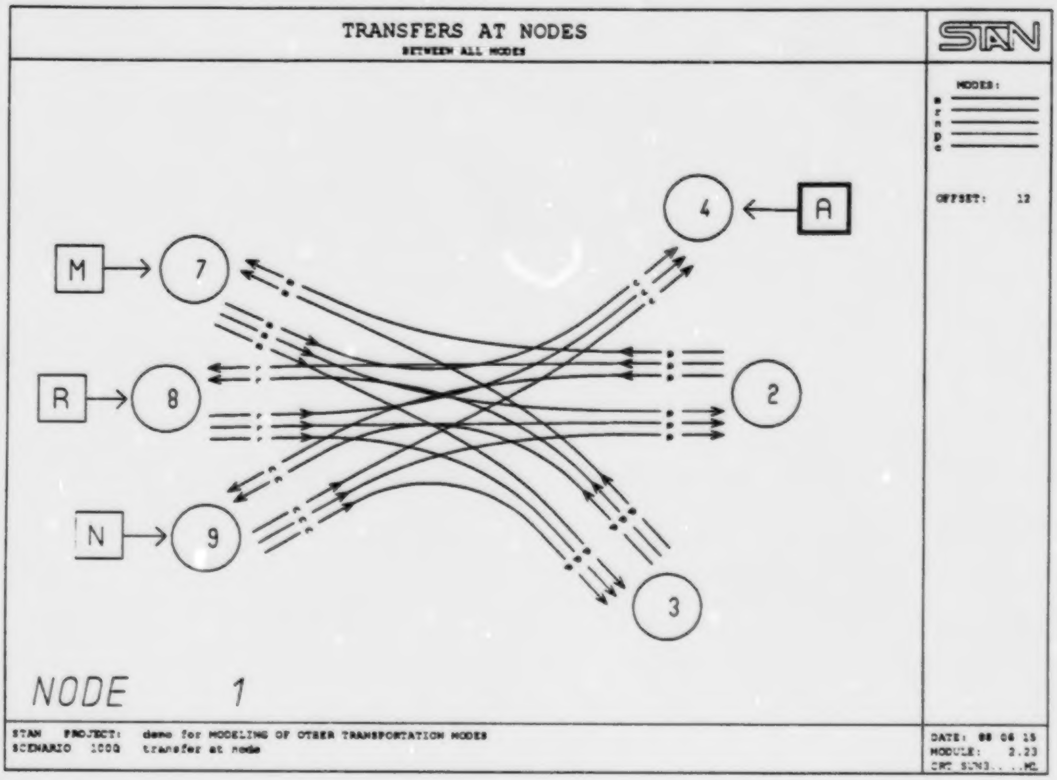

TRANSFERS AT NODES
BETWEEN ALL NODES
STAN
NODES:
OFFSET: 12
A
4
M
7
R
8
2
N
9
3
NODE 1
STAN PROJECT: demo for MODELING OF OTHER TRANSPORTATION MODES
SCENARIO 1000 transfer at node
DATE: 88 06 15
MODULE: 2.23

Figure 5 Transfers at Node 1 of the Port Network

At node 1, transfers

$$(\mathrm{M},1) \leftrightarrow (1,2) \quad \text{and} \quad (\mathrm{M},1) \leftrightarrow (1,3)$$

$$(\mathrm{R},1) \leftrightarrow (1,2) \quad \text{and} \quad (\mathrm{R},1) \leftrightarrow (1,3)$$

$$(\mathrm{N},1) \leftrightarrow (1,2) \quad \text{and} \quad (\mathrm{N},1) \leftrightarrow (1,3)$$

represent the loading (unloading) of freight in (to) vehicles and its movements through the port's facilities, while transfers $(\mathrm{M},1) \rightarrow (1,\mathrm{A})$, $(\mathrm{R},1) \rightarrow (1,\mathrm{A})$ and $(\mathrm{N},1) \rightarrow (1,\mathrm{A})$

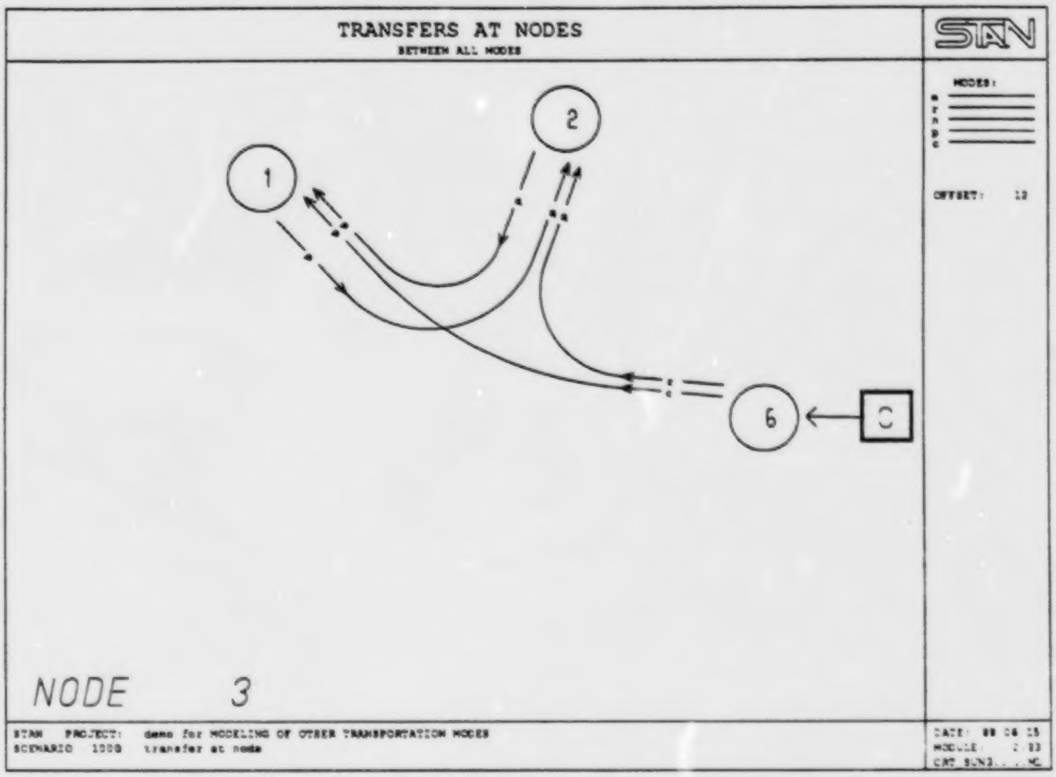

TRANSFERS AT NODES
BETWEEN ALL MODES
STAN
MODES:
m
r
h
p
c
OFFSET: 12
1
2
6
C
NODE 3
STAN PROJECT: demo for MODELING OF OTHER TRANSPORTATION MODES
SCENARIO 1000 transfer at node
DATE: 88 06 15
MODULE: 2.33
CRT SUN3... NC

Figure 6 Transfers at Node 3 of the Port Network

permit the excess flow to reach its destination. Cost and delay functions, that we discuss below, are then used to route the flows and to account for their economic and temporal aspects.

At node 1, a number of movements are, explicitly or implicitly, prohibited. Thus, transfers between non-port modes (such as (M,1) ↔ (R,1), for example) are simply not specified. On the other hand, transfers between links of the port mode, such as (2,1) → (1,A) and (1,2) ↔ (1,3), are prohibited when necessary, by using suitably high valued (theoretically infinite) delay functions. Similarly, at node 3, the definition of the transfer (C,3) → (3,2) allows the departure of the flow from the centroid, while the absence of the transfer (C,3) → (3,1) effectively forbids this movement.

All port links and transfers must have cost, delay and energy consumption functions associated with them in order to compute the generalized cost in the assignment procedure. Cost and energy consumption data may be found out from data recorded and used by port authorities. The definition of delay functions is, however, a more difficult undertaking that requires careful analysis.

Delay functions play essentially two roles: first, to account for the time that goods spend waiting, either in vehicles or in warehouses, for resources (loading/unloading equipment, ships, vehicles, etc.) to become available; second, and as a consequence of the first, to route the flow to the appropriate link when resources are not available. In both cases, delays reflect the characteristics of the facility and the volume of traffic that passes through it, and aim to capture the congestion effects when volumes are high as compared to the capacity of the facility. In STAN the delay functions are volume/delay relations which must be convex or monotone increasing.

Loading and unloading equipment usually has known, fixed productivity rates stated as the number of tons of a given product handled per hour. This service rate may be combined with arrival rates deduced from the traffic volumes and, by using appropriate queueing models, average waiting time functions may be derived. Polynomial delay functions may then be fitted to these functions by using the techniques described in Chapter 7. When the information needed for the calibration of delay functions by such a procedure is not available, simple linear functions based on historical data, may be used as well. These functions may be product specific and are associated with transfers.

In order to define delay functions that will route the traffic on the links of the port network according to the available port and warehouse capacity, a different approach is needed. A port normally has a specific handling capacity for a given product during a given period. This capacity is mainly derived from the storage available and the capacities of the vehicles disposable to move the product out of the port. The product flows generally incur a "linear" utilization (handling, waiting, etc.) cost up to this capacity, while all flows that exceed it cannot be handled immediately.

Since the strategic multi-mode multi-product transportation model is a static formulation where short term capacity effects may not be modeled, this is not, however, a feasible approach in STAN. To overcome this difficulty, these linear relations with strict capacity restrictions are transformed into convex volume/delay functions (the techniques discussed in Chapter 8 may be used for this purpose) and then, *discharge* links are added to receive the traffic that exceeds the capacity. Link (1,A) is the archetypical discharge arc, although link (1,3) is in the same class. It is then sufficient to calibrate a delay function for the discharge link that is higher for volumes that are less than the capacity, and lower for volumes exceeding it. As the assignment is performed using a minimum cost criteria, the discharge link will not be used before the capacity is reached. Figure 7 illustrates this concept. Note that this approach may be generalized to three functions, where the third takes the excess flow of the combined capacity of the first two. In all cases, the calibration of these functions is a rather delicate procedure.

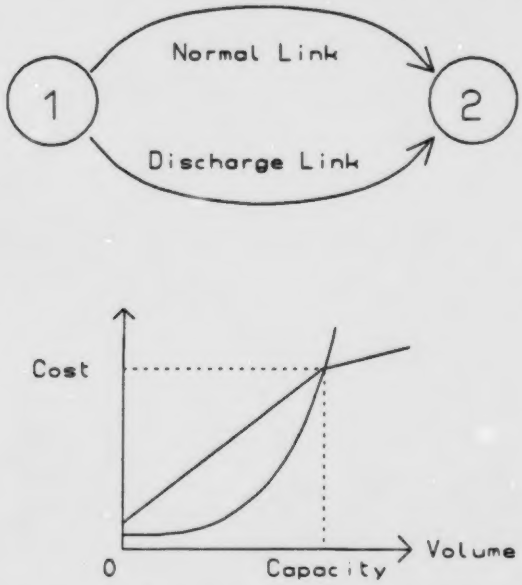


Figure 7 Discharge Link and Port Delay Function

5.2.4 Utilization modes

Warehousing is an essential port activity. Furthermore, warehousing is essentially a temporal activity. It is thus rather awkward to integrate it into a long term static model that uses extended planning periods. The use of discharge arcs permits, however, to adequately address a large range of warehousing situations.

When the flow passing through a port is less than its capacity, the so called *in-period* warehousing activities are captured by the discharge arcs, for both import and export flows, regardless if the origin/destination is in port (centroid A) or outside it (centroid B). For export flows, link (1,3) serves as discharge arc for link (1,2), while link (2,3) plays the same role for link (2,1) for import flows.

It is also relatively easy to model the situation when the export flows exceed the capacity of the port. When the destination of the flow is the port (centroid A), link (1,A) represents the discharge arc for the combined capacity of links (1,2) and (1,3). The flow moving through this link is then considered to wait for the next period, before being shipped out of centroid C. It is called *inter-period* warehousing.

When the destination is a port outside the country (centroid B), one does not know a priori the volume of traffic that will use a particular port. The same approach may, however, be used if either a new discharge arc (1,B) (similar to link (1,A)), or a link (A,B) with a zero delay is introduced into the network. The second alternative is preferable, since it achieves the same result while simplifying the calibration of the delay functions for the discharge arcs. In all cases, the volume flowing through the port is again split between the direct (arc (1,2)) and the indirect (link (1,3)) ship loading.

For import flows, however, a similar approach can not be efficiently applied. In order to model inter-period storage, discharge arcs must be added from all ports (from centroids A, plus arcs (B,A)) to all other centroids in the network. This would add too many arcs to the network and the corresponding computing effort would not be justified by the expected gain in model accuracy. In order to overcome this problem, an approach consisting in a preprocessing of the data may be used: If the origin is the port, then the total demand should be compared with the capacity of the port and the demand matrices should be modified accordingly. When the origin is outside the country, the demand should be compared to the capacity of all the ports that might be used, as one does not know before hand which ones will be used, nor in which proportions. The demand should then again be ajusted to fit the available capacity.

It is relevant to note that, even if the port has sufficient capacity to handle the traffic, the rest of the network may not, particularly when some commodity is confined to a limitted number of transportation modes. Thus, it may be always worthwile to use the maxflow procedure (see Chapter 6) before an assignment, in order to estimate the traffic that may actually pass through the transportation network.

We conclude this section with the remark that we have not explicitly and separately considered in our port modeling the various, dedicated or general, terminals that form a port. This would normally bring about a level of detail incompatible with the representation of the general transportation system. As all link and transfer functions may be calibrated individually for each product, the aggregation of all terminals in one network yields an adequate strategic model of a port's operations. It is always possible, if absolutely necessary, to consider several terminals, or aggregations of terminals, for a given port, by linking together an appropriate number of port networks. Caution should then be exercised to appropriately define the demand matrices and the connections to the port centroids and with the rest of the transportation network. The general way to link the ports to the transportation system is the subject of the following section.

5.3 NAVIGATION MODES

Regions that have maritime or ocean access, or a significant lake or river system, depend on navigation as an important transportation means. As it is valuable to the moving, over long distances, of bulky, palettizable and containerizable products, transportation by ship is vital to the international trade and, therefore, to a nation's import and export activities. On the other hand, in some regions rivers offer the only efficient means of reaching isolated setlements, while coastal navigation may often supply an energy efficient alternative for the movement of goods between different regions. We have therefore included water based transportation modes in the STAN modeling approach and this section is dedicated to their presentation.

One usually differentiates navigation types according to the body of water upon which the moving takes place. Here, *ocean* navigation indicates the long course navigation, traversing seas and oceans and linking the region to ports outside the country. *Coastal* navigation represents movements, along the coast, between domestic ports. *Internal* navigation takes place on the rivers and lakes of the region.

The usual restriction imposed by the strategic level of representation and analysis are, however, also present for the water transportation modes. We propose therefore a general model of the navigation mode that respects these requirements and that may be adapted for all navigation types.

The navigation network consists of the same elements as the rest of the transportation network: regular nodes, centroids and directional, modal links. One should note that the main role of the navigation network is to connect the various ports in and outside the region. In general, the navigation network may not be linked directly with the other modes: a port is necessary.

A possible approach to modeling the connection of ports to the navigation mode is by using the existing information on the *navigation lines* that operate along the coast of the region. Figure 8 illustrates this approach, where one line calls at ports 1, 2 and 3, while another calls at ports 1 and 3.

Navigation lines are akin to train services and the reasons that have lead to the choice of not to consider rail services are equally valid for these navigation services. Planning for service lines requires a level of detail that is not compatible with that of strategic planning. In most cases, this is in the domain of tactical planning, where

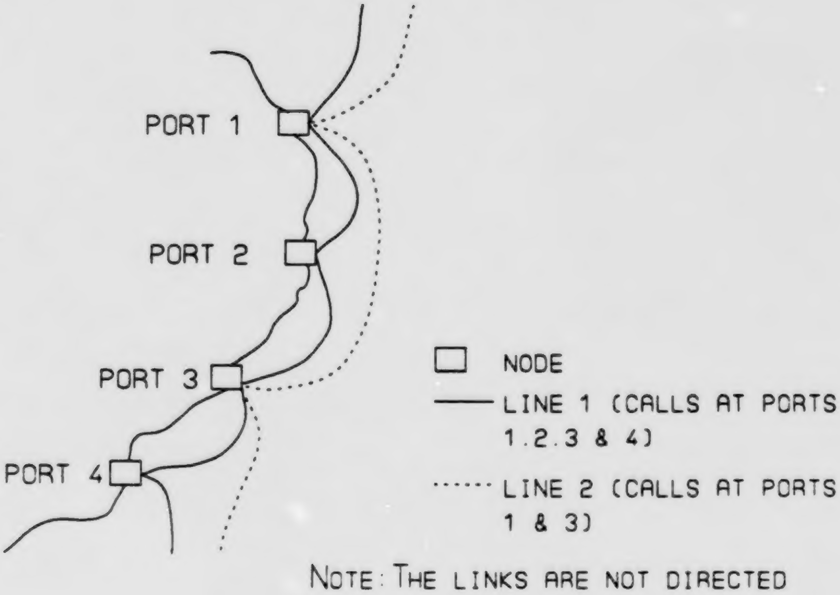

PORT 1
PORT 2
PORT 3
PORT 4
NODE
LINE 1 (CALLS AT PORTS 1.2.3 & 4)
LINE 2 (CALLS AT PORTS 1 & 3)
NOTE: THE LINKS ARE NOT DIRECTED

Figure 8 Navigation Network - Line Representation

the design and frequency of the service lines have to be determined. In addition, the description of actual service lines in a network that may be modified for a large variety of prospective scenarios implies either that one knows well the future routes of the navigation lines, or that the navigation network will not change significantly in the proposed scenarios.

Therefore, a schematic representation of the navigation system was chosen. It is illustrated in Figure 9. One regular node, refered to as the *navigation* node, is added to each port. Then, links of the *navigation mode* are used to sequentially connect all these nodes, and each one of them to the corresponding port subnetwork.

When modeling sea movements, the navigation mode corresponds to coastal traffic. For internal navigation, the network may generally use the same representation.

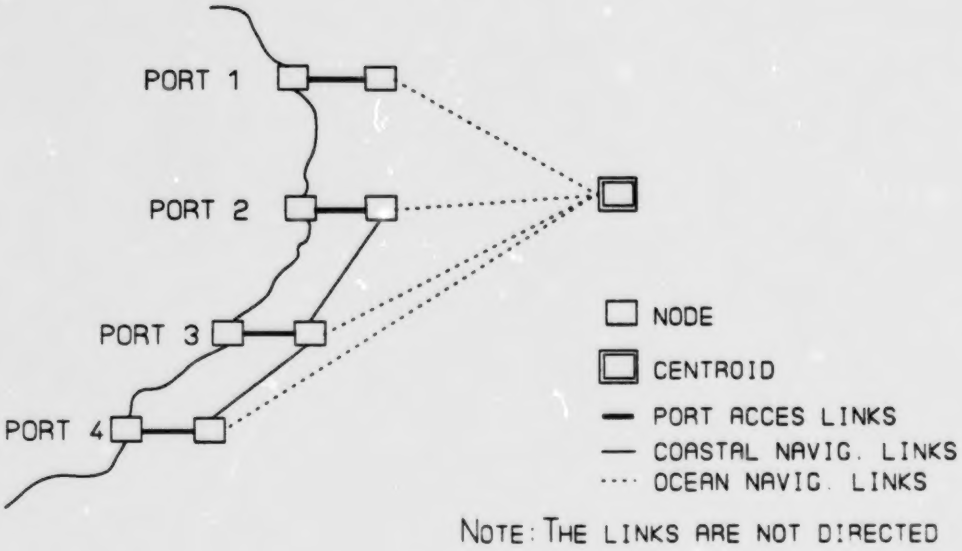

PORT 1
PORT 2
PORT 3
PORT 4
NODE
CENTROID
PORT ACCES LINKS
COASTAL NAVIG. LINKS
OCEAN NAVIG. LINKS
NOTE: THE LINKS ARE NOT DIRECTED

Figure 9 Navigation Network

Naturally, when modeling navigation on a lake, direct, crossing links may be added, where appropriate.

For ocean navigation, the modeling approach is different. As before, navigation lines cannot be explicitly considered. Furthermore, even our simplified approach may became too cumbersome if the whole terrestrial globe is to be considered, as all possible foreign ports may appear in the network. This would generate an excessively large network and would also significantly complicate the determination of the various link and transfer attributes. For example, since a significant part of the import and export traffic is usually carried by ships under foreign pavilion, vehicle characteristics, as well as link capacities and costs, may vary considerably and be quite difficult to estimate. In fact, this level of detail is not at all required.

In most national strategic analyses, the emphasis is on the means of moving the products through the internal transportation network to the ports, in order to have the export products ready for departure and the imported products shipped inland to their final destinations. This observation is corroborated by the fact that quite often the origin/destination transportation demand matrices have ports as both origins and destinations. We have thus chosen to represent the external world by a single centroid with connectors to the navigation node of each port. Note that this centroid is centroid B in the port representation of Section 5.2.

We illustrate these notions in Figure 10, where three ports are linked to the coastal navigation network and to the external centroid. For the clarity of the presentation, the discharge arcs, (A,B), have not been indicated for each port.

Vehicle characteristics and link attributes have to be defined in the usual way. For the navigation mode, Chapter 4, Part IV, illustrates how this was done for the Brazilian application. One word of caution, however, concerning the capacity associated with the navigation network. Since navigation is quite energy efficient, capacities have to be associated (via volume/delay functions) with the navigation links to prevent unrealistic flows. Navigation capacities may usually be computed, for each product and for each pair of origin/destination ports, from the frequency and type of ship used on each navigation line to transport this product. Given the requirements of our network representation, however, capacities cannot be specifically assigned in this way to each origin/destination. A more global value, representing the combined capacity of the navigation system, has to be used instead for some links of the navigation subnetwork. The links that interconnect the navigation nodes are the most appropriate elements to which to associate this capacity. The delay functions of the links connecting the navigation nodes to their corresponding ports are better employed to model the queues that arriving ships encounter at the entries to the ports.

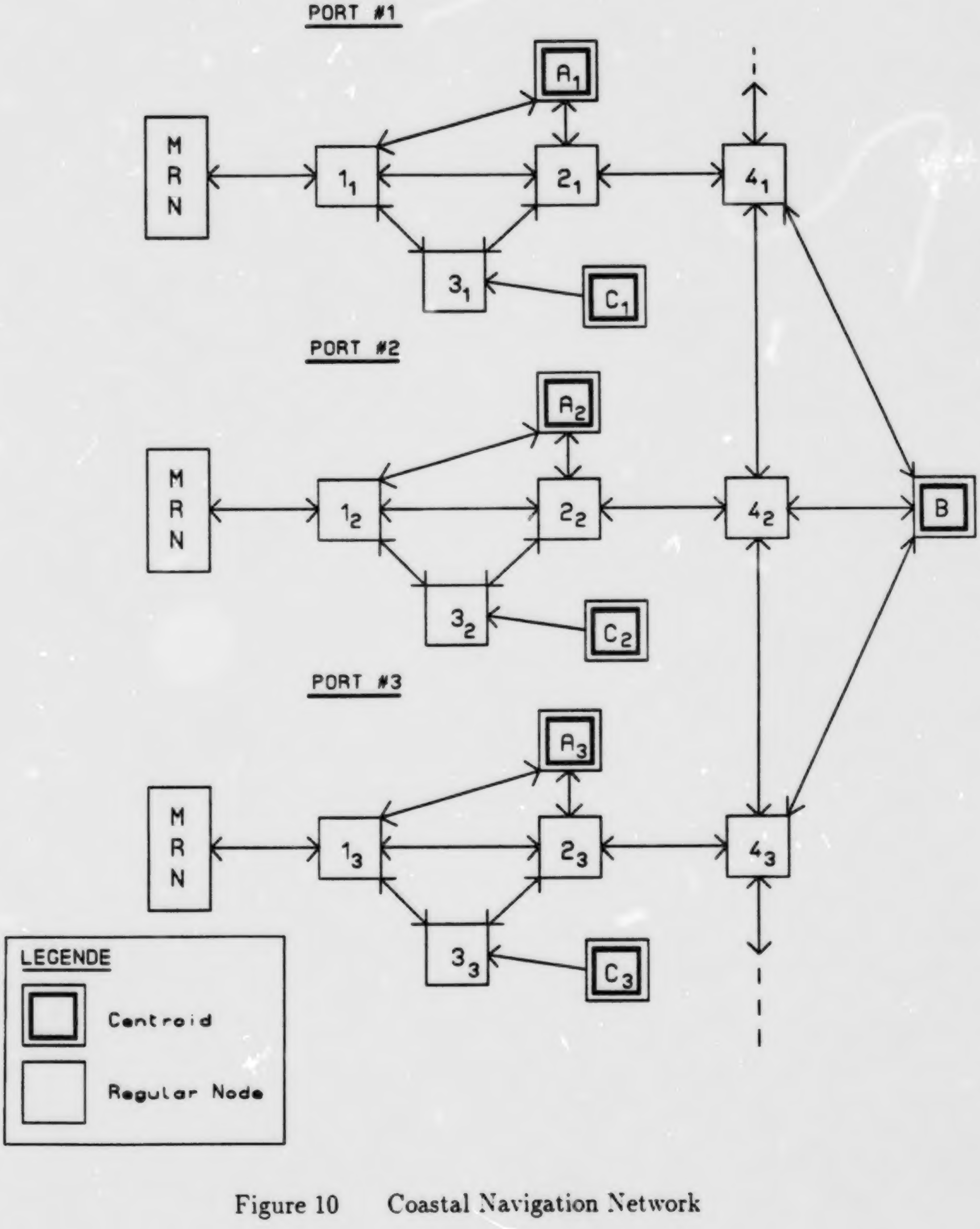

PORT #1
PORT #2
PORT #3
MRN
1_1
2_1
3_1
4_1
A_1
C_1
1_2
2_2
3_2
4_2
A_2
C_2
B
1_3
2_3
3_3
4_3
A_3
C_3
LEGENDE
Centroid
Regular Node

Figure 10 Coastal Navigation Network

5.4 GENERAL CONSIDERATIONS

The general modeling principles used for the representation of the rail, road, navigation and port modes are equally valid for the modeling of any other transportation mode. We want, however, to briefly emphasize some important factors that have to be considered when the representation of a mode is contemplated for inclusion in the STAN modeling framework.

Any modal subnetwork is exclusively made of centroids, nodes and links. While centroids and nodes may be shared by several modes, links are mode specific. Since products are assigned on specified subsets of permited modes and since permited transfers in STAN have to be explicitely defined, particular attention has to be dedicated to the inter and intra modal connectivity of the networks.

In particular, if a product has to be exclusively assigned to one mode, such as products moving on pipelines, the corresponding modal subnetwork must be connected. This implies that for each origin/destination pair with a positive traffic demand to be moved on the subnetwork, a path must exist from the origin centroid, through the modal network, to the destination centroid. On the other hand, the appropriate transfers have to be explicitly defined if a product needs to use a multi-modal subnetwork.

Modes represent a very powerful modeling tool. One must try, however, not to abuse it. Thus, one has to resist the temptation to represent services through modes. Other than the point that service considerations at a national level are more adequately addressed at the tactical level of planning, the multiplication of modes exacerbates the network connectivity issue and also requires the definition of an increased number of transfers. Consequently, the global performance of the assignment procedure is affected without achieving a real gain in the quality of the results.

Cost, energy consumption and delay functions have to be associated to every link and transfer of a modal subnetwork and these functions may be product specific. Delay functions, in particular, are most important as they are used to capture the congestion effects due to limited capacities and high traffic volumes. In STAN, all these functions must be convex or, at least, monotone increasing. The polynomial approximation method of Chapter 4, or the penalty approach of Chapter 8 may be used to obtain delay functions that include capacity considerations and that respect the conditions of connexity or monotonicity.

When contemplating the specific components of the cost, delay and energy consumption functions associated to the modal links and transfers, foremost consideration should be given to the uniform treatment of all the modes included in the specific STAN scenario.

Finally, it is worthwhile to recall that all traffic in STAN is measured in tons. Consequently, "typical" vehicles for each product for each mode have to be defined. The characteristics of these vehicles should aim to reflect the most used real vehicle or combination of vehicles. When vehicles may be assembled into convoys, such as trains for the rail mode, the characteristics of these convoys may be specified in STAN. The evaluation of the time delays and transportation costs on links and transfers, may thus be greatly improved by using the corresponding number of vehicles and convoys, rather than by estimating these costs per tonnage of product transported.

5.5 BIBLIOGRAPHY

Rico, N. (1986), *Contributions à l'analyse stratégique du transport de marchandises: calcul du flot maximal multiproduit et modélisation portuaire,* M. Sc. Thesis, Département d'informatique et recherche opérationnelle, Publication #509, Centre de recherche sur les transports, Université de Montréal.

Wilmes, P. and Frankel, E. (1973), *Port Analysis an Planing,* in *Proceedings of the First International Conference on Transportation Research,* Bruges, Belgium, Transportation Research Forum, 1973.

CHAPTER SIX

A NOTE ON MAXIMAL MULTICOMMODITY NETWORK FLOWS

6.1 INTRODUCTION

The purpose of this note is to propose an algorithm for the solution of the maximum multicommodity network flow problem in a capacitated network, based on an Augmented Lagrangean reformulation of the problem. As we shall show, this way of considering this problem requires the solution of a sequence of minimum convex cost multicommodity flow problems which have the same form as a network equilibrium problem with variable demand, for which algorithms have been developed that can solve large scale problems efficiently. Previous approaches to solve this problem resort to price directive decomposition methods for large scale linear programs (Ford and Fulkerson (1958)) or resource directive decomposition methods for large scale linear programs (Grinold (1968), Held et al (1974), Kennington and Shakley (1976)). A heuristic method for solving the problem has been suggested by Gratzer and Steiglitz (1972) based on notions derived from a resource directive decomposition method. Exact methods for solving this problem are known only for the case of two commodity maximal flow (Rothschild and Whinston (1966) and Sakarovitch (1973)).

6.2 THE PROBLEM

We consider a network (N, A) where N is the set of nodes and A is the set of directed arcs. The nodes n, $n \in N$, represent origins, destinations and intersections of links; the directed arcs a, $a \in A$, represent the transportation facilities. The flows on link a is v_a and the unknown maximal flows for origin/destination pairs i, $i \in I \subset N \times N$, are g_i. The origin to destination flows g_i may use directed paths k, $k \in K_i$, and the path flows are denoted h_k.

The maximum multicommodity network flow problem in a capacited network, where each arc is subject to a capacity constraint c_a, may be formulated as

$$\max \sum_{i \in I} g_i \tag{1}$$

$$\text{subject to} \sum_{k \in K_i} h_k - g_i = 0,\ i \in I \tag{2}$$

$$h_k \geq 0,\ k \in K_i,\ i \in I \tag{3}$$

$$v_a \leq c_a,\ a \in A \tag{4}$$

$$g_i \geq 0,\ i \in I \tag{5}$$

$$v_a = \sum_{i \in I} \sum_{k \in K_i} \delta_{ak} \cdot h_k,\ a \in A \tag{6}$$

$$\text{where } \delta_{ak} = \begin{cases} 1 & \text{if link } a \text{ belongs to path } k \\ 0 & \text{otherwise} \end{cases}$$

The problem has been formulated in the space of path flows in order to simplify the statement of the conservation of flow equations, as in (2). The nonnegativity constraints on the path flows (3) ensure that $v_a \geq 0$, $a \in A$.

In this formulation, a distinct commodity is associated with each distinct origin/destination pair. If two or more commodities are associated with some origin/destination pairs, then an additional index p would be needed to denote the commodities, e.g. g_i^p.

6.3 SOLUTION BY THE AUGMENTED LAGRANGEAN METHOD

The reformulation of the problem (1) - (5) for solution by the Augmented Lagrangean method consists in dualizing constraints (4), which results in the following problem (see Bertsekas (1982))

$$\min_{v,g} \max_{\lambda \geq 0} L_U(v, g, \lambda) = -\sum_{i \in I} g_i + \frac{1}{2U} \sum_{a \in A} \left\{ max(0;\ \lambda_a + U(v_a - c_a))^2 - \lambda_a^2 \right\} \quad (7)$$

Subject to (2), (3), (5), (6)

where U is the penalty constant and λ_a, $a \in A$, are the Lagrange multipliers associated with each constraint in (4).

The solution of this problem is obtained by the following general algorithm:

Step 0 (Initialisation).

$l := 0;\ \lambda_{a,i} := 0,\ a \in A$ where l is an iteration index

U_0 is an initial value of the penalty constant.

Step 1 (Solve problem in v, g for fixed λ).

$$\min_{v,g} -\sum_{i \in I} g_i + \frac{1}{2U_l} \sum_{a \in A} \left\{ max(0;\ \lambda_{a,l} + U_l(v_a - c_a))^2 - \lambda_{a,l}^2 \right\} \quad (8)$$

Subject to (2), (3), (5), (6)

Let $(v_{a,l}; g_{i,l})$ be the solution of this problem.

Step 2 (Adjustment of multipliers).

$\lambda_{a,l+1} := max(0;\ \lambda_{a,l} + U_l(v_a - c_a)),\ a \in A.$

Step 3 (Optimality Test).

If $\| \lambda_l - \lambda_{l+1} \| \leq \varepsilon$, STOP.

Otherwise continue to

Step 4 (Adjustment of penalty constant).

Set U_{l+1} such that $U_{l+1} \geq U_l$

$l := l+1$ and return to Step 1.

The subproblem to be solved in Step 1 is a convex cost multicommodity network flow problem which appears at first to be difficult, however a closer inspection reveals that it has the same structure as a network equilibrium problem with variable demand. In order to do so we recall its formulation.

The network equilibrium problem with variable demand seeks to compute the origin to destination demands g_i wich are given by monotone decreasing demand functions $G_i(u_i)$, $i \in I$, where u_i is the least cost of travel for origin/destination pair i, and the arc flows v_a, $a \in A$, where the cost of traveling on a link $s_a(v_a)$, $a \in A$, is a monotone increasing function. This problem, as is well known, is formulated as:

$$\min -\sum_{i\in I} \int_0^{g_i} W_i(y)dy + \sum_{a\in A} \int_0^{v_a} s_a(x)dx \tag{9}$$

Subject to (2), (3), (5), (6)

where $W_i(g_i)$ is the inverse of the demand function $G_i(u_i)$.

If we juxtapose the corresponding terms of (8) and (9) it is possible to observe that $-\sum_{i\in I} g_i$ may be rewritten as $-\sum_{i\in I} \int_0^{g_i} dy$ and that

$$\sum_{a\in A} \left\{ max(0;\ \lambda_{a,l} + U_l(v_a - c_a))^2 - \lambda_{a,l}^2 \right\}$$

is a convex function of v_a, which may be stated as $\sum_{a\in A} S_a(v_a)$, as defined by the expression in the curly brackets. Thus (7) may be rewritten as

$$\min_{v,g} -\sum_{i\in I}\int_0^{g_i} dy + \sum_{a\in A} S_a(v_a) \tag{10}$$

and the similarity between the objectives (8) and (9) are evident as

$$\tilde{S}_a(v_a) = \int_0^{v_a} s_a(x)dx$$

is a convex function.

The purpose of this exercise is to conclude that (8) is a rather easy problem to solve, for which a plethora of first order solution algorithms have been developed (see Florian (1986)), the most common being the linear approximation algorithm of Frank and Wolfe (1956) and its variants. It is important to note that (8) may not be solved by second order methods, since the objective function is only once differentiable. Furthermore, relatively large scale problems (~2000 - 4000 nodes, 6000 - 12000 arcs) may be solve in reasonable computation times. It is interesting to note that given this observation, (8) has the structure of a network equilibrium problem where the inverse demand functions are constant and equal to 1.

6.4 SOME COMMENTS

The approach suggested in this note is suitable for computing maximum multicommodity flows for large networks where the arc capacities do not necessarily represent strict constraints.

From the computational viewpoint, if a primal algorithm is used in Step 1 to solve (7), the solution from iteration l is feasible for iteration $l+1$ since only the objective function is modified in steps 2 and 4. Thus, at iteration $l+1$ only a reoptimization of (8) is required where $(v_{a,l}; g_{i,l})$ serve as an initial feasible solution . In an eventual implementation of this algorithm, the particular choices made for values of the penalty constant U and other possible stopping criteria that may be used, would need careful attention.

It is also interesting to draw some conclusions regarding the uniqueness of the solution from this way of reformulating the maximum multicommodity flow problem. While it is known that the solution of a linear program is not necessarily unique, by inspecting (10) it is easy to draw stronger immediate conclusions. Since the coefficient of each variable g_i, $i \in I$ is constant, it follows that the demands g_i are not unique: in fact the only variables which are uniques are the arc flows v_a which equal to the capacity c_a in optimal solutions. The consequence is that this model is best used to find the "bottleneck" arcs in a multicommodity network and care must be exercised to intepret the optimal values of the demands g_i, $i \in I$.

This approach to solving for maximal multicommodity flows may be easily extended to consider lower bounds on the demands g_i, $i \in I$ and "weights" γ_i, $\gamma_i > 0$, which would give preference to one product over others (i.e. max $\sum_{i \in I} \gamma_i g_i$).

Finally, it is sometimes revealing to inspect the Kuhn-Tucker conditions of a convex optimization problem subject to linear constraints. These are, for (10) subject to (2), (3), (5), (6):

$$\left.\begin{aligned} \sum_{a \in A} \delta_{ak} S_a'(v_a) - u_i - \gamma_k &= 0 \\ h_k \cdot \gamma_k &= 0 \\ \gamma_k &\geq 0 \end{aligned}\right\} \; k \in k_i, \; i \in I \qquad (11)$$

$$\text{and} \qquad -1 + u_i = 0 \qquad i \in I \qquad (12)$$

where u_i, γ_k are the dual variables corresponding to (2), (3) respectively. For each path h_k such that $h_k > 0$, it follows, by using (11) and (12), that the "length" of the path, $\sum_{a \in A} \delta_{ak} S'_a(v_a) = 1$ whatever the values of U and λ_a, $a \in A$.

6.5 COMPUTATIONAL EXPERIENCE

The multicommodity maxflow algorithm presented in the previous sections was implemented in a code written in Fortran '77 and executed on SUN/3 workstations (MC68020 equipped with MC68881 floating point unit operating at 15 Mhz under UNIX). The variable demand network equilibrium problem, that is defined for fixed λ, was solved by using the PARTAN variant of the Linear Approximation Method (Florian, Guélat and Speiss (1987)). A small sample problem was solved. Table 1 below gives the results for this problem which is given in Figure 1.

Tolerence (ϵ) for outer iteration	.05	.05	.05	.05	.05	.05	.05
Tolerence (ϵ) for subproblem	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}
Penalty U	2.00	1.75	1.50	1.25	1.00	0.50	0.05
Execution Time (seconds)	6.26	6.52	10.44	6.50	6.66	5.70	5.54
Value of objective fonction	159.9995	160.0002	160.0002	160.0065	160.00058	160.00019	160.000137
O/D flows	40.13 39.98 39.92 39.98	40.07 39.99 39.96 39.99	40.07 39.99 39.94 39.99	40.14 39.98 39.90 39.98	40.33 39.95 39.77 39.95	40.18 39.94 39.92 39.94	47.66 37.67 37.13 37.67

Table 1

(13 nodes, 19 links, 4 centroids, 5 products)

The same problem was solved as a linear program (115 variables, 179 constraints) by using the XMP code. The time required for the execution was 5.54 seconds and the value of the objective fuction was 160.0000.

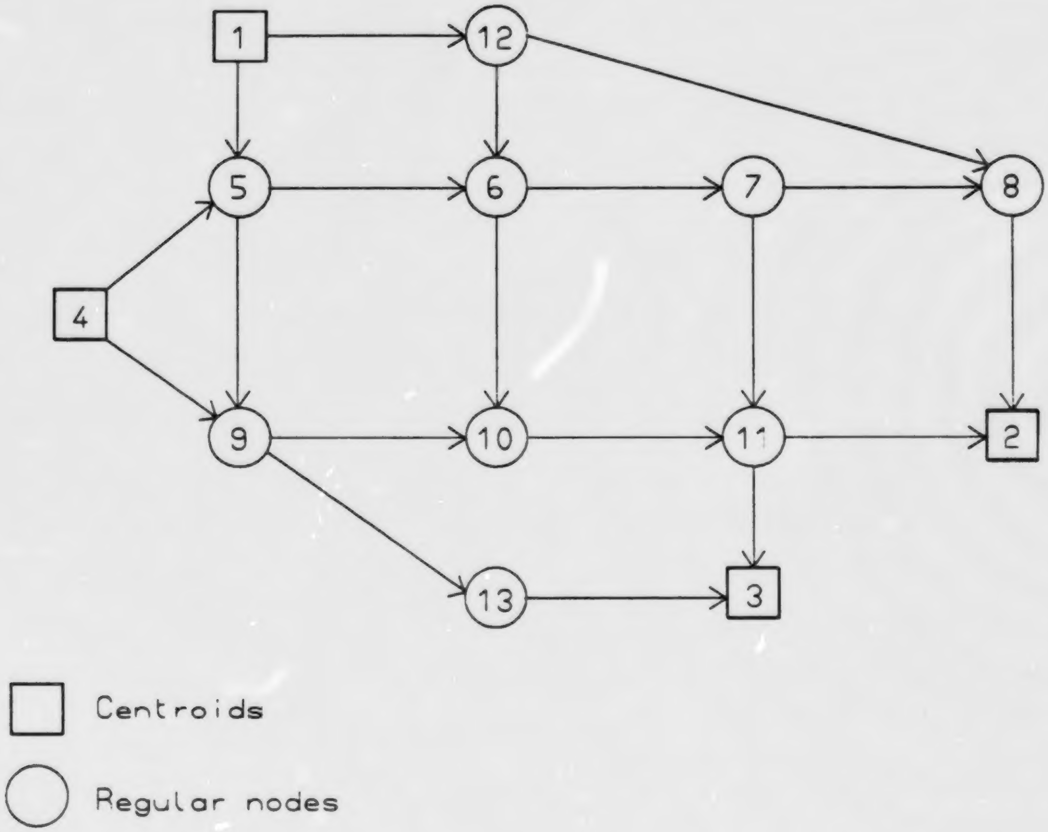

1
12
5
6
7
8
4
9
10
11
2
13
3
Centroids
Regular nodes

Figure 1 Test network

These results indicate that the approach is competitive, even for small size problems, where the nonlinear programming approach is expected to be inferior to that of solving small linear programs.

6.6 CONCLUSION

We have showed in this note that the maximum multicommodity flow problem may be solved by a sequence of convex cost multicommodity flow problems, by using an Augmented Lagragean reformulation of the problem. In particular the convex cost problems that must be solved have the same structure as the network equilibrium problem with variable demand, which is a relatively easy problem to solve. Thus, one may solve for maximum multicommodity flows in a capacited network without having to resort to primal or dual decomposition methods of linear programming. The computational result obtained with a preliminary implementation of the algorithm indicate that this approach is practical for applications which arise in analysing the capacity "bottleneck" of freight transportation networks.

6.7 BIBLIOGRAPHY

Bertsekas, D.P., *Constrained Optimization and Lagrange Multipliers Methods*, Academic Press, New York (1982).

Florian, M., "Nonlinear Cost Network Models in Transportation Analysis", *Mathematical Programming Study* 26, (1986), 167-196.

Ford, L.R. and Fulkerson, D.R., "A Suggested Computation for Maximal Multicommodity Network Flows", *Management Science* 5, (1985), 97-101.

Frank, M. and Wolfe, P., "An Algorithm for Quadratic Programming", *Naval Research Logistics Quarterly* 3, (1965), 95-110.

Gratzer, F.S. and Steiglitz, K., " A Heuristique Approach to Large Multicommodity Flow Problems", *Proceedings of the Symposium on Computer Communications Networks and Teletraffic*, J. Fox (editor) (1972), 311-324.

Grinold, R.C., "A multicommodity Max-Flow Algorithm", *Operational research* 16. (1968), 1234-1238.

Held, M., Wolfe, P. and Crowder, H.P., "Validation of Subgradient Optimization", *Mathematical Programming* 6, (1974), 62-68.

Kennington, J.L. and Shakley, M., " An Effective Subgradient Procedure for Minimal Cost Multicommodity Flow Problems", Tech. Report IEOR 750010, Dept. of Ind. Eng. and Oper. Res., Southern Methodist University (1976).

Rothschild, B. and Whinston, A., "On Two Commodity Network Flows", *Operations Research* 14, (1966), 1121-1129.

Sakarovitch, M., "Two-Commodity Network Flows Linear Programming", *Mathematical Programming* 4, (1973), 1-20.

CHAPTER SEVEN

CALIBRATION OF DELAY FUNCTIONS

In rail transportation, the train travel time between two consecutive stations depends not only on the distance to be travelled and on the average speed of the train, but also on the physical characteristics (signalization, number of sidings, single or double track, etc.) of the line and on its utilization as measured by the total number of trains that pass on the line during a given period of time.

In order to model this travel time, it is possible to develop *over-the-line delay* formulae that determine the *average train travel time* on a rail line segment as a function of the characteristics and *capacity* of the line and the *total train traffic* (Crainic, 1987) . Such formulae may be used at the operational level to forecast and analyse the performance of specific rail lines; similar but more aggregated formulae are a particularly important component of network optimization models of freight rail transportation systems at tactical (medium term) or strategic (long term) planning levels.

STAN is build around a multi-mode multi-product network optimization model aimed at the strategic analysis and planning of national freight transportation systems. The model, described in Chapter 3, incorporates over-the-line delay functions that are used in the global objective function to represent the impact of rail line capacity and congestion on the performance of the whole freight transportation system.

Due to its multi-mode environment and strategic planning objectives, a number of approximations had to be made when building the network model. Among these, the most critical for an adequate representation of the rail mode are the estimation formulae for the total train traffic on each link and the polynomial approximation of the over-the-line delays.

The purpose of this chapter is to present and analyse procedures that may be used for the calibration of these formulae, especially for the links of a large network.

We want to emphasize that the following discussion, procedures and general calibration approach are equally valid for all delay functions of the same general class and shape as those presented in this Chapter. In particular, the methodology applies to the yard put-through delay functions described in Chapter 4 . We focus on over-the-line delay functions, however, because, by sheer numbers alone (a "few" yards compared to hundreds or even thousands of links in most realistic national applications), links and link characteristics play a central role in STAN and the strategic planning of national freight transportation systems.

The report is organized as follows. We first present the formulae used in the modeling of the rail mode, followed by the description of the data used for the experiments. The next section is dedicated to the presentation and analysis of the calibration procedure for the polynomial over-the-line delay function. The link train flow approximation formula is then discussed. The Appendix contains the descriptions and instructions for using the various procedures that may be helpful in performing the computations implied by the calibration of the delay functions for the links of a large network.

7.1 NOTATION AND BASIC FORMULAE

The general structure of the national freight network model and the requirements of the solution algorithm used place a number of restrictions and guide lines on the possible model formulations for each of the transportation modes that it considers, among which the most important are:

a) Only product flows, expressed in some measure of weight, may be defined on the network;
b) No explicit capacity constraints may be included in the model;
c) The total cost objective function must be convex.

For the rail mode, these hypotheses have two main consequences. First, as some important components of a rail transportation system with path-like structures (e.g. train services and frequencies, train and car flows) cannot be defined on the network, product flows must be transformed into an equivalent number of cars and trains for each link. These approximations are subsequently used in computing link delays and costs. Secondly, the link or over-the-line delays have to reflect the delays due to the train congestion and to enforce the capacity restrictions of the line.

A number of additional hypotheses were inspired by the context of the brazilian transportation system and by the needs of strategic planning:

a) Passenger trains are not considered on the network;
b) All freight trains have equal priority;
c) On a link, all trains moving in the same direction, travel at the same (and constant) speed;
d) Consequently, no more than two trains meet at the same time;
e) On each link, the traffic is balanced, meaning that the number of trains travelling in each direction is approximately the same.

A detailed presentation of the freight rail model based on these hypotheses appears in Chapter 4 . Here, we only summarize the main results.

The number of trains travelling on a particular link a is computed by the following formula:

$$t_a = \sum_p \frac{(v_a^p + (v_a^p/w_p) * vw_p)}{(\alpha_p * \beta_a)}$$

where,

t_a : number of trains on link a;

v_a^p : quantity of product p travelling on link a (weight units);

w_p : quantity of product p loaded into a vehicle p (weight units);

vw_p : weight of an empty vehicle p (weight units);

α_p : weight of a train of product p (weight units);

β_p : link correcting factor.

The average over-the-line delay for a train travelling on a single track link may be computed, consistent with the assumptions made and given the capacity of the link and the train flows in each direction, by a formula proposed by Petersen (1974):

$$D_a = \frac{FRT_a * CAP_a + (FRT_{\bar{a}} * t_{\bar{a}} - FRT_a * t_a)}{CAP_a - (t_{\bar{a}} + t_a)}$$

where the two links a and $\bar{a}$ represent the two directions of travel on a single track rail line and where,

D_a : delay on link a;

CAP_a : capacity of link a;

FRT_a : free running time on link a;

$FRT_{\bar{a}}$: free running time on link $\bar{a}$;

t_a : train flow on link a;

$t_{\bar{a}}$: train flow on link $\bar{a}$.

We identified several problems related to the above formula:

a) The asymptotic (toward infinity) behavior of the delay when the total train flow approaches the link capacity (see Figure 1);

b) When fully developed, by substituting the equation for the number of trains, the formula consists of too many terms, implying a more difficult user interface, as well as longer computation times for the function evaluation procedure of the assignment algorithm;

c) The convexity of an objective function that includes this delay formula cannot be demonstrated in a general case.

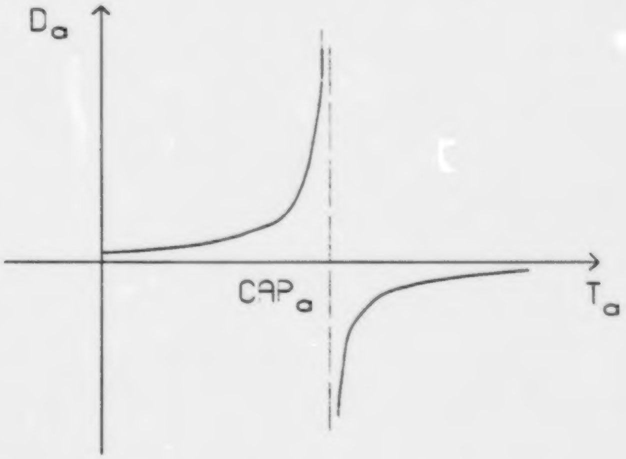


Figure 1 General Shape of a Delay Function

We have therefore proposed a polynomial approximation of the delay function:

$$\bar{D}_a = FRT_a * (1 + \phi_1 * T_a + \phi_2 * (T_a/CAP'_a)^{\exp})$$

where,

$\bar{D}_a$: estimate (approximation) of delay on link a;

T_a : total train flow on the link $T_a = t_a + t_{\bar{a}}$

CAP'_a : value of the total train flow where the line begins to be considered congested;

ϕ_1 : linear parameter (to be calibrated);

ϕ_2 : nonlinear (congestion) parameter (to be calibrated);

exp : linear parameter (to be calibrated).

The parameters may be calibrated by non-linear regression methods and we present results of this procedure later on. Simply used however, this approach is not entirely satisfactory for real applications, since too large a number of different polynomials would be generated (in fact, a different polynomial for each link of the network!). Also, the degree of the polynomials generated by non-linear regression methods are, generally, quite high, increasing the solution time of the assignment algorithm.

As an alternative, a heuristic procedure may be used to calibrate the parameters of the polynomial approximation. The specific objective of this chapter is to present the details of this heuristic approach, to examine its efficiency and goodness of fit and to explore its utilization to generate a limited number of "accurate" delay formulae for the links of a large network. First, we briefly present the data used for the experiments.

7.2 DATA DESCRIPTION

Our experiments were conducted with data from the Canadian National Railways (CNR). It consists of traffic demand forecasts for the western part of CNR's network. Once aggregated for tactical planning purposes, it represents a network of 107 nodes and 95 links of which 86 are single track links. Traffic demand is defined for 25 aggregated products, while two freight train types (normal and rapid), as well as passenger trains, are allowed on the network.

A tactical planning network model (Crainic, Ferland and Rousseau, 1984) has been applied with this data in order to obtain train service routings and frequencies, yard classification workloads and traffic itineraries through the service (trains and yards) network. Detailed over-the-line delay formulae are used in this model (Crainic. 1982, 1987).

For each single track link, the following data is available:

- origin and destination nodes;
- length;
- free running time;
- capacity (estimated) per week;
- number of passenger trains (both directions);
- traffic volumes per product (both directions);
- number of freight trains of each train type (both directions);
- average over-the-line delay for each train type.

A constant speed of 45 miles/h was assumed for all links, which simplified the delay formula to:

$$D_a = \frac{FRT_a * (CAP_a + t_{\bar{a}} - t_a)}{CAP_a - t_{\bar{a}} - t_a}$$

As the line capacities were not readily available, being implicitly considered by the over-the-line delay formula, Colson's formula was used to estimate the link capacity for this experiment:

$$CAP_a = \frac{K * (2 * 1440 * 7)}{S_a^2 + \phi}$$

where,

S_a : siding to siding time;

ϕ : signalization time;

K : percentage of effective capacity, accounting for the maintenance of the line.

We used two minutes signalization time, 85% effective capacity and a week as planning period.

A sample of 10 links was extracted from the network, with the objective of selecting various combinations of link capacity and length. Table 1 displays the sample and its characteristics.

In an exploratory approach, that is as much art as it is science, our initial objective is to calibrate a polynomial delay function for each link in the sample. We then try to use these polynomials to generate one polynomial delay function which adequately fits all links or, at least, a few polynomials each of which adequately fits a subset of links.

link number	link	FRT	CAP
1	(5,15)	293,33	212,92
2	(17,18)	18,67	193,57
3	(19,23)	398,67	72,75
4	(24,31)	562,67	123,73
5	(26,31)	489,33	48,42
6	(45,46)	52,00	223,69
7	(49,50)	72,00	261,35
8	(59,65)	161,33	146,57
9	(70,73)	168,00	237,72
10	(104,105)	52,00	25,44

Table 1 Link Sample

7.3 CALIBRATION PROCEDURE

This section is dedicated to the presentation of the calibration procedure for the parameters of the polynomial over-the-line delay function. It begins with a general statement of the procedure, followed by the presentation of the results of the numerical experiments performed using the sample data and the conclusions that we draw concerning the calibration of each one of the parameters. Based on these conclusions, we suggest possible approaches for the efficient calibration of a limited number of polynomial delay functions for the links of a realistically large network. The last part of this section presents a validation of the polynomial functions calibrated by our procedure by comparing their goodness of fit with that of the polynomials obtained by non-linear regresssion methods.

Note that when the appropriate function is introduced in its first step, the calibration procedure may be used to approximate any delay function of the general form illustrated in Figure 2, inclusive, in particular, of yard delay functions.

7.3.1 The procedure

The procedure estimates values for the three parameters, exp, ϕ_1 and ϕ_2, of the polynomial delay function $\bar{D}_a(\phi_1, \phi_2, \text{exp}, T_a)$, such that it fits closely to the delay $\bar{D}_a(t_a, t_{\bar{a}})$ computed by a given formula. Here, Peterson's delay formula (1974) is used.

For link a , the general procedure is:

Step 1. Generate a set of points (T_a, D_a) = (total train traffic, delay), $T_a = t_a + t_{\bar{a}}$, for various combinations of train flows t_a and $t_{\bar{a}}$. Fix the value of CAP'_a at the level of the total train flow where the link begins to be considered congested.

Step 2. Plot the points obtained in Step 1, "observe" the curvature of the scatter diagram and fix **exp** accordingly.

Step 3. Estimate ϕ_2 by fitting

$$D_a = FRT_a * (1 + \phi_2 * (T_a/CAP'_a)^{\text{exp}})$$

for high values of T_a, i.e. $T_a \geq CAP'_a$.

Step 4. Evaluate ϕ_1 to adjust

$$D_a = FRT_a * (1 + \phi_1 * T_a^* + \phi_2 * (T_a^*/CAP_a')^{\text{exp}})$$

for a point in the linear part of the scatter diagram, e.g. $T_a^* \in [0,1 * CAP_a \, , \, 0,2 * CAP_a]$, and the value of ϕ_2 calculated at Step 3.

Step 5. Repeat Steps 2 to 4 for "intelligent" choices of **exp** and choose the best fit polynomial.

To judge the goodness of fit of the polynomials, we use the *mean (average) relative error* (ARE) and the *maximum relative error* (MRE) as computed by the differences between the two delay formulae evaluated at the same points.

Note that all numerical computations required by this procedure may be performed, "simultaneously" on all the links of a given link set, by using the computer programs presented in the Appendix. As to the "educated" guesses that are also a necessary part of the procedure, the experiments, and their analysis, that we present in the rest of the section reveal much of the essence of choices made.

7.3.2 Numerical experiments

The sets of points (T_a, D_a) were first generated for all the links in the sample. The procedure considers all integer values for the total train flow, $T_a = t_a + t_{\bar{a}}$, between 0 and 92% of each link's capacity. For *even* total train flow values, only one delay evaluation is performed, supposing equal train flows in each direction. For *odd* values, two delays are computed: one using $t_a = (T_a + 1)/2$ and $t_{\bar{a}} = (T_a - 1)/2$, and another using the reverse values. The average of these two delays is then used. Figure 2 displays the plot of these points for link number 1. Note that, since link 1 is as representative as any other link in the sample and since we want to keep the presentation simple, only plots relative to link 1 are illustrated in this Chapter.

The choice of the degree of the polynomial, **exp** , is guided by a dual criterion. On one hand, as good a fit as possible is desired for the polynomial. This usually means, since it improves the fit in the "congested" part of the plot, a high degree. On the other hand, the inclusion of high degree polynomials in the objective function of the STAN model may induce numerical problems when a multi-mode assignment is performed. In particular, these problems may seriously affect the procedure that numerically computes the derivatives of the function and are especially important in high congestion situations when flows are near capacity. The convergence and precision of the algorithm may then be affected. Or, on one hand, congestion in the yards and on the lines of a rail transportation system is usually quite high, and, on the other hand, the STAN assignment algorithm requests a very large number of evaluations of both functions and theirs derivatives. One would then prefer a polynomial degree as low as possible. A compromise has to be reached.

For the links of our sample, we tried **exp** equal to 5, 6 and 7, values which seemed reasonable for the observed plots (of which Figure 2 is an example) and for an efficient assignment algorithm. The results are displayed in Table 2, while Figure 3 illustrates them for link 1.

One notes that polynomials of degree 6 and 7 are significantly better than polynomials of degree 5. The difference is much less significant, however, when comparing polynomials of degree 6 and 7 only. Therefore, consistent with the previous discussion, we choose a degree 6 for our delay functions.

The proper estimation of ϕ_2 and, consequently, that of the polynomial, depends upon the right choice of CAP_a'. This value indicates, as a certain percentage of the

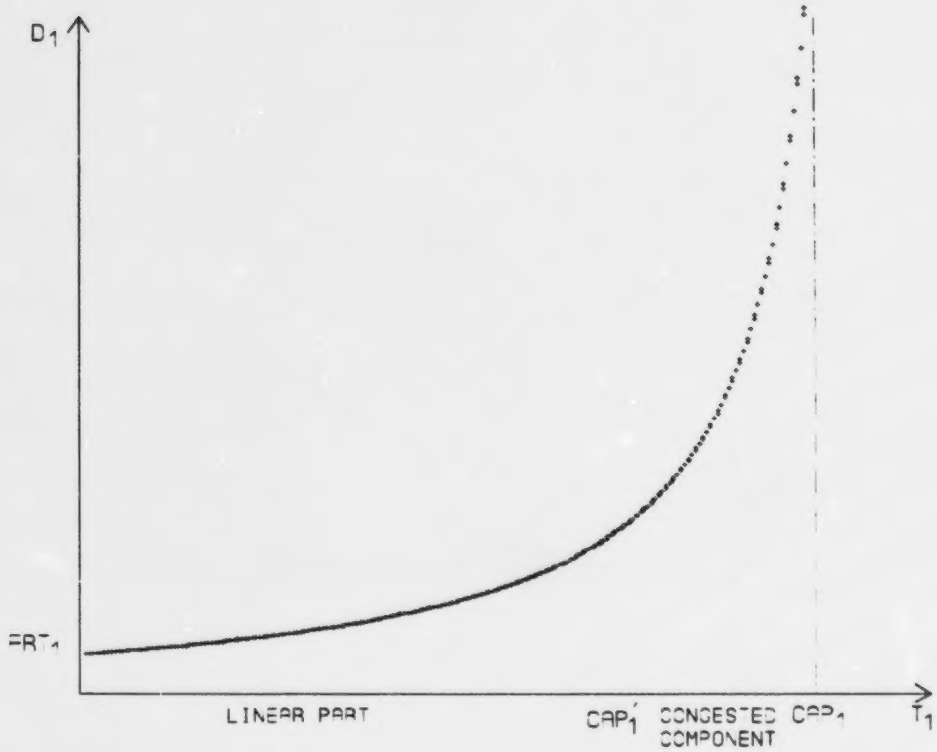


Figure 2 Plot of the Points (T_a, D_a) for Link 1

link capacity, the threshold where the link begins to be considered congested. It is also used to indicate in the calibration procedure, the point where the two separate calibrations (i.e. of ϕ_2 in the congested part of the function and of ϕ_1 in its linear part) meet and are linked.

CAP_a' should therefore be chosen at an "equilibrium" point between the linear and congested components of the delay function (see Figure 2). Choosing it too low would imply a poor fit of the congested part of the polynomial. On the other hand, too high a value for CAP_a' generates unsatisfactory results for the linear component of the polynomial approximation.

Link Number	exp = 5		exp = 6		exp = 7	
	ARE	MRE	ARE	MRE	ARE	MRE
1	8,16	27,00	7,19	20,57	7,98	17,51
2	8,31	27,48	7,25	20,90	7,98	17,69
3	7,68	25,31	7,00	19,54	7,96	17,08
4	8,08	26,73	7,17	20,44	8,00	17,52
5	7,83	25,52	7,12	19,54	8,09	17,02
6	8,19	27,10	7,20	20,64	7,98	17,53
7	8,21	27,15	7,20	20,68	7,97	17,57
8	7,97	26,24	7,14	20,13	8.00	17.37
9	8,16	26,92	7,20	20,48	7,99	17.40
10	7.98	26.69	7.25	20.78	8,21	18.09

Table 2 Fitness Characteristics for Polynomials of Various Degrees

These observations are illustrated by the figures of Table 3 showing the *mean relative errors* and *maximum relative errors* of several polynomial approximations of the over-the-line delay for link 1, obtained with varying congestion thresholds. Figure 4 displays some of these polynomials. Consequently, a value of $CAP'_a = 80\% * CAP_a$ was deemed satisfactory for the links of our sample. Note that this 80% value for the congestion threshold is in agreement with previous results (Roy, 1986) obtained by using a different link sample.

It is also noteworthy that, in Step 3 of the calibration procedure, an alternate rule may be used to evaluate ϕ_2:

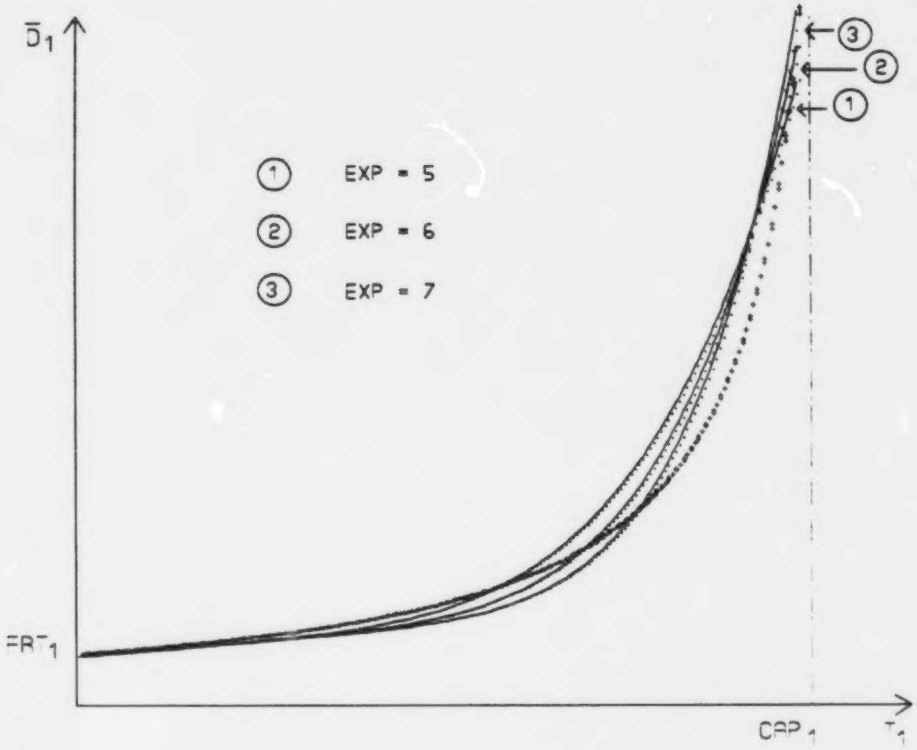


Figure 3 Polynomial Delays of Various Degrees

STEP 3'. Estimate ϕ_2 by fitting $D_a = FRT_a * (\phi_2 * (T_a/CAP'_a)^{\text{exp}})$
for high values of T_a, i.e. $T_a \geq CAP'_a$.

Table 3 displays the results obtained for link 1 using the two rules, 3 and 3' respectively. As can be seen, rule 3 consistently outperforms rule 3', and this observation is true for all the links of the sample. This behaviour may be explained by the fact that rule 3 makes use of existing information that the alternate rule discards: FRT_a, the "fixed cost" of traveling on link a. Or, using the "fixed cost" information in Step 3 of the procedure has the effect of decreasing the value of ϕ_2 and, consequently, the relative weight of the non-linear component of the polynomial approximation.

Graphically, this corresponds to a "lowering" of all the congested area toward the linear part of the plot. Consequently, the subsequent fitting of the linear compoment and the linking of the two separate calibrations are improved, thus improving the overall fit of the polynomial. Consequently, we recommend the use of rule 3 in the calibration procedure.

Congestion	Rule 3'		Rule 3	
Threshold	ARE	MRE	ARE	MRE
0,40%			29,99	109,20
0,50%	44,37	159,81	20,75	75,67
0,60%	21,93	80,02	11,43	39,67
0,70%	13,18	46,76	7,90	24,22
0,76%	10,95	37,66	7,22	20,76
0,80%	10.25	34,74	7,19	20,57
0,85%	10.29	34.91	7,70	23,25

Table 3   Fitness Characteristics, Varying Congestion Thresholds and Calibration Rules

Using $CAP'_a = 80\% * CAP_a$, $T^*_a = 0,1 * CAP_a$ and **exp** equal to 6, we obtained polynomial delay functions, one for each link in the sample; their parameters and corresponding goodness of fit are displayed in Table 4. These results are extremely satisfactory.

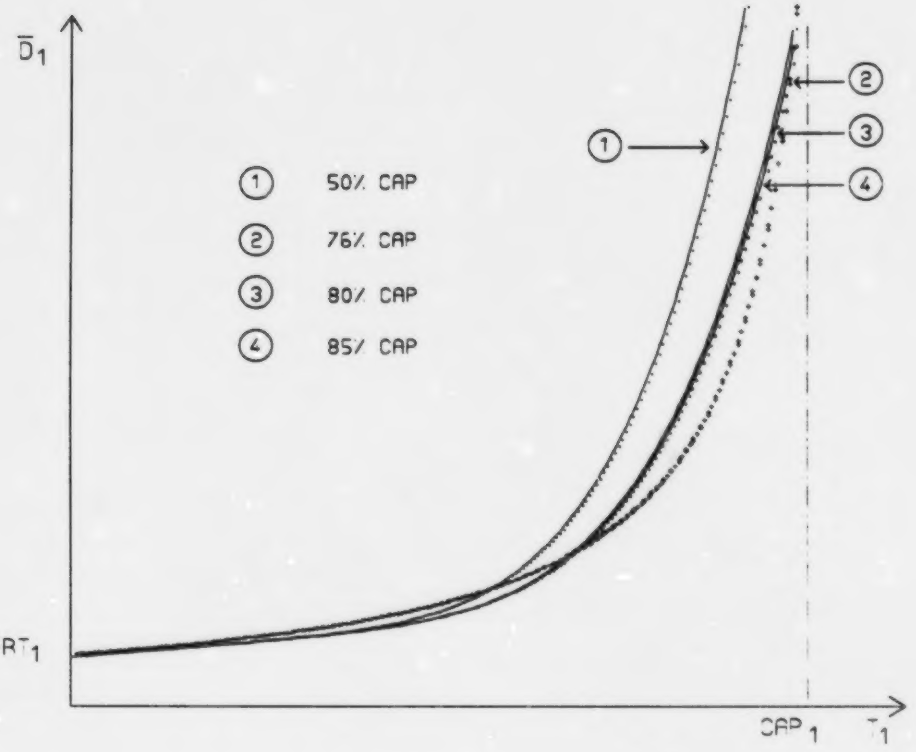


Figure 4 Polynomial Delays, Varing Congestion Thresholds

7.3.3 Utilization Modes

The calibration procedure, used as described in the previous section, generates good polynomial approximations for the over-the-line delays for the links of an arbitrarily large network. It generates, in fact, **one** function for each link, and these functions are likely to be different for each link. For STAN and most applications of realistic dimensions, this is not acceptable. We therefore present in this section how to use the same procedure to calibrate a limited number of "good" polynomial functions for the links of a realistically large network.

Link Number	ϕ_2	ϕ_1	1,1/CAP	ARE	MRE
1	4,127	0,00521	0,00517	7,19	20,57
2	4,143	0,00573	0,00568	7,25	20,90
3	4,079	0,01521	0,01512	7,00	19,54
4	4,122	0,00895	0,00889	7,17	20,44
5	4,096	0,02251	0,02272	7,12	19,54
6	4,130	0,00496	0,00492	7,20	20,64
7	4,131	0,00425	0,00421	7,20	20,68
8	4,108	0,00754	0,00750	7,14	20,13
9	4,124	0,00466	0,00463	7,20	20,48
10	4,163	0,04266	0,04324	7,25	20,78

Table 4 Polynomial Over-the-Line Delays

First, let us examine the polynomials generated at the end of the previous section for the links of our set. Two main conclusions may be drawn from the figures displayed in Table 4:

a) ϕ_2 presents very similar values for all links;

b) Due to the chosen value for T_a^*, ϕ_1 is always very close to 1,1/CAP
which would suggest that capacity is a good criterion for the grouping of links.

We may use then the same polynomial for all the links of the network. In the case of our sample data, using exp=6, $\phi_1 = 1,1/CAP$ and ϕ_2=4,12252, the average of the values of ϕ_2 for all links, we obtained the results displayed in Table 5.

Link number	ARE	MRE
1	7,236	20,341
2	7,239	20,339
3	7,192	20,330
4	7,223	20,338
5	7,135	20,229
6	7,237	20,345
7	7,239	20,344
8	7,226	20,337
9	7,239	20,340
10	6,989	20,138

Table 5 Fitness of the "Average" Polynomial

This method may, however, generate quite significant errors for large networks, since there may be links with very different characteristics of length, capacity, traffic, etc. It may be interesting then to group links into classes and calibrate, following the preceding method, one polynomial for each of these classes. Considering the previous remark on the value of ϕ_1 and the known relationships between capacity, congestion and delay, classes should be defined according primarily to the link capacity and, secondly, to other link characteristics, such as the free running time.

For our example, four link classes may be formed:

A: large capacity, links 1,2,6,7 and 9;
B: medium capacity, links 4 and 8;
C: small capacity, links 3 and 5;
D: very small capacity, link 10.

For each class, the parameters of the polynomial delay and the corresponding fitness characteristics are shown in Table 6, where the values of ϕ_1 and ϕ_2 for each class are the averages over the values obtained for the polynomial of the links in that particular class.

This procedure gives generally better results than the previous one and is preferable, in particular when the network contains links of significantly different types.

We note, finally, that all these procedures may be used even when the network is totally or partially hypothetical, as in the case of planned modifications to the infrastructre. It is sufficient to specify the corresponding values for the link characteristics, especially capacity and free running time, and the calibration procedure may be applied.

Class	exp	ϕ_1	ϕ_2	ARE	MRE
A	6	0.00492	4.12	7.625	19.495
B	6	0.00819	4.12	7.893	18.986
C	6	0.01892	4.12	6.517	24.513
D	6	0.04324	4.12	6.978	20.086

Table 6 Polynomial Delays for each Class

7.3.4 Validation - regression procedures

The parameters of the approximation polynomial for the over-the-line delay function may also be estimated directly, for a given link, by non-linear regression methods. This approach could, at least theoretically, provide us with more accurate approximations than the heuristic method. Yet, for reasons that will became clear by the end of this section, we do not retain this approach. We do, however, compare the two methods to validate the results of the heuristic calibration procedure.

Our experiments were performed using the non-linear regression program AR from the BMDP package on four links representative of their respective link class: links number 1, 4, 3 and 10.

Initially, the parameters of the four polynomials were calibrated using the regression program without any constraint. The results are presented in Table 7.

Link	exp	ϕ_1	ϕ_2	ARE	MRE
1	8,96	2,4882	0,009215	6,014	11,797
4	8,92	2,4968	0,015815	5,950	11,699
3	8,45	2,5800	0,026000	5,330	10,508
10	8,93	2,5174	0,076680	5,752	11,577

Table 7 Unconstrained Non-Linear Regression

Since the degrees of the polynomials estimated by the unconstrained non-linear regression procedure are too high for our requirements, we reran the regression calibration procedure with the degree of the polynomial fixed at six. Table 8 presents the results of this experiment, while Table 9 displays the corresponding figures for the heuristic calibration procedure. Figure 5 displays the polynomials, for link 1, that correspond to these three methods.

Link	exp	ϕ_1	ϕ_2	ARE	MRE
1	6	3,56347	0,00558	5,02	14,92
4	6	3,55450	0,00965	4,94	13,66
3	6	3,42230	0,01743	4,17	12,54
10	6	3,67862	0,04442	5,64	11,22

Table 8 Non-Linear Regression with Constrained Degree

Several conclusions may be drawn from an examination of these results, which we summarize as follows:

a) The polynomial functions obtained by using non-linear regression methods, either constrained in the degree of the polynomial or not, do not offer a significantly better fit than those obtained by the heuristic calibration procedure. The *average relative errors* are similar in all cases. As for the *maximum relative errors* they are indeed lower for the regression methods but, the MRE being an extreme mesure, they are not an adequate criterion in this case, as can be observed in the plots of Figure 5.

b) The two non-linear regression methods result in polynomial functions of pratically similar quality, as measured by their goodness of fit.

c) The polynomial degree generated by the unconstrained regression method is generally too high for the desired efficiency of the assignment algorithm, as discussed earlier.

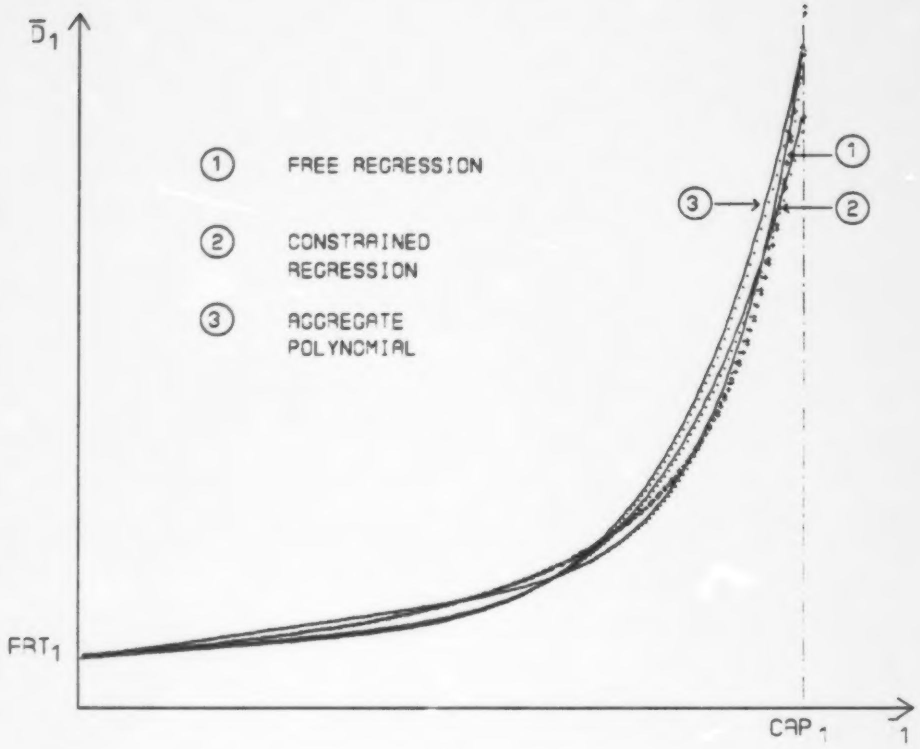


Figure 5 Aggregate and Non-Linear Regression Polynomials

Link	exp	ϕ_1	ϕ_2	ARE	MRE
1	6	4,12	0,00492	7,625	19,495
4	6	4,12	0,00819	7,893	18,986
3	6	4,12	0,01892	6,517	24,513
10	6	4,12	0,04324	6,978	20,086

Table 9 Heuristic Calibration Procedure

One also notes that the regression approach still requires the execution of Step 1 (delay data generation) as well as most of the data manipulation of our procedure.

On the other hand, it offers less possibilities for user control and aggregation when a limited number of polynomials have to be selected for a large number of links.

It is thus quite clear that the non-linear regression approach is not entirely satisfactory for our function fitting problem; nevertheless, it permitted us to validate a heuristic procedure which offers extremely satisfatory results and an adequate degree of flexibility and adaptibility.

7.4 TRAIN AND DELAY APPROXIMATION

This section is dedicated to a brief analysis of the train flow approximation formula. We also briefly examine the behavior of the train and delay approximations when the unique train type hypothesis is relaxed.

We have seen, when presenting the modeling of the rail mode (Chapter 4) that there are but two methods to approximate, at the link level, the complex process that determines, at the network level, the train composition, length and frequency: an *average* formula and a *by product* approach. Given the *typical car* defined, for each product p, by its dead weight vw_p and its average load w_p, and given the link flows v_a^p, one approximates the *number of cars* n_a^p of product p travelling on a particular link a by

$$n_a^p = \frac{v_a^p}{w_p}$$

and the *average* formula gives then a total link flow t_a of

$$t_a = \sum_p \frac{n_a}{n}$$

for an average number of cars per train n .

The *by product* approach, used in STAN, computes the number of trains travelling on a link by

$$t_a = \sum_p \frac{v_a^p + (v_a^p/w_p) * vw_p}{\alpha_p * \beta_a}$$

where α_p represents the weight of a train made-up entirely of cars of product p while β_a is the link correcting factor that ajusts the approximated train flow to the particular features (maximum load, signalization, etc.) of the link.

We have already identified the undeniable conceptual and modeling advantages of the second formula. We now present some figures to numerically back-up our analysis.

Table 10 displays the values of the train traffic, on each direction of each link, for the transportation plan generated by the tactical model (Crainic, Ferland and Rousseau, 1984) on the network from which the sample data was extracted (the *Planned* columns) and for the two approximation formulas (the *By product* and the *Average* columns, respectivelly). Two estimations of the α_p were used for the *by product* approximation and will be discussed momentarily. Meanwhile, we focus our attention on the figures in the *Minimum* columns. Values of $\beta_a = 1$ and $n = 100$ are assumed throughout the experiments. Note that the *Planned* entries were computed in integer arithmetic, while all other figures were computed in real arithmetic and were subsequently rounded off.

An examination of these figures shows that reasonable approximations may be obtained for the train link flows by using the *average* and the *by product* formulae. It also clearly shows that the second approach systematically and significantly outperforms the *average* method.

The fit is less good on links where, in the tactical transportation plan, a relatively small amount of traffic is moved by a number of trains significantly different from the number normally suggested by the car and product flows. This situation usually occurs on secondary lines and is generated either by the particular characteristics of the line, or by local consideration of service criteria. A judicious use of the link correcting factor β_a will allow, however, to calibrate the *by product* formula to obtain an excellent fit even in these cases. This is evidently not true of the simple average method, which again demonstrates the superiority of our approach.

Link Number	Planned		By product				Average	
			Minimum α		Maximum α		No α	
	a	$\bar{a}$	a	$\bar{a}$	a	$\bar{a}$	a	$\bar{a}$
1	30	30	27	27	12	11	24	24
2	48	48	47	46	25	23	42	41
3	7	7	3	5	1	2	3	4
4	50	48	49	49	27	34	43	43
5	37	38	40	38	21	26	35	36
6	3	3	3	3	2	2	2	2
7	5	5	6	6	2	3	5	6
8	11	10	11	9	5	4	9	8
9	4	6	3	5	2	2	3	4
10	6	6	3	3	2	1	3	3

Table 10 Planned and Estimated Train Link Flows

Our experiments, as illustrated in the figures of Table 10, have shown that some attention should be paid to the estimation of the α_p, the *by product train weights*. The difficulty here lies in obtaining figures which give a good calibration of the train link flow formula for all the links of the network, given that each product is generally transported between several origin-destination pairs. As discussed in previous chapters, several products may be represented in the model by the same aggregated "real" commodity. This would allow a more "personalized" calibration of the corresponding α_p on groups of origin-destination pairs with similar loading characteristics. It would be utopic to believe, however, that one could realistically consider each origin-destination-commodity combination as a separate product. Hence, the need to calibrate the factor α_p for several origin-destinations pairs for each product.

We then examined, for each product in the sample, the distribution of its α_p factor over the origin-destination pairs between which it is transported and Table 11 displays the average and extreme values of these distributions.

These figures illustrate our observation that, for most products moved through a transportation system, the average car load for a given product generally presents small variations among the various origin-destination pairs: for most products in the sample the difference between the maximum and minimum figures is relatively small, while the average value for the car load is very close to the minimum load figure. The utilization in this case of the minimum of the distribution, instead of the mean, will generate a higher number of trains and, thus, have a correcting effect on the tendency of all approximation formulae based, even loosely, on averages, to underevaluate the actual figures.

The case of product 99 illustrates the other situation, when there are significant variations among the origin-destination pairs for the same product. For each origin-destination pair, commodity 99 is the aggregation of all products for which the shipment size is under a certain tonnage threshold. Weighted averages were used to compute the various product characteristics. This aggregation generated large variations in the characteristics of product 99 among origin-destination pairs. For this type of product, better results are obtained if the corresponding α_p is measured where the car load figures for most origin-destination pairs are concentrated: as the median of the distribution, instead of its minimum.

Product	Average	Maximum	Minimum
1	119,8	150	110
5	90,0	90	90
6	129,1	140	125
7	134,1	150	125
8	130,0	130	130
13	120,0	120	120
15	108,3	140	75
16	130,0	130	130
17	74,5	80	70
18	65,1	70	60
19	70.0	70	70
20	70,0	70	70
21	108,6	130	90
22	93.6	100	85
23	94,3	100	75
24	100,0	100	100
25	87,8	100	85
26	94,4	100	95
27	80,9	100	65
28	83,5	100	75
29	28,4	40	25
30	57,3	60	50
31	78,8	90	75
32	60,4	70	60
99	56.5	150	50

Table 11 Average and Extremes of the α_p Distributions

(Hundreds of Tons)

We suggest therefore the following procedure to calibrate the train load factor α_p for a given product p :

1) Determine the train load factor for each origin-destination pair implicated in the transportation of the product and consider the resulting distribution of the α_p;
2) If the loading is homogeneous, i.e. small difference between the minimum and maximum values of the distribution and the mean is close to the minimum, then fix the value of the α_p factor equal to the minimum train load for that product;
3) Else (heterogeneous loading), fix the α_p equal to the median of the distribution for the particular product.

We used this procedure to calibrate the α_p for the train approximation formula and generated the figures appearing in the columns *By product - Minimum* of Table 10. As already noted, these figures are better than those obtained by the *average* approach that corresponds to the utilization of a unique α factor for all products. Table 10 also displays the figures corresponding to the α_p fixed to the maximum values of the distributions. This option is clearly non satisfactory.

Before concluding, we briefly examine the question of the overall precision of our approximations, especially when the hypothesis of a unique freight train type and no passenger trains on the network is not verified.

Table 12 presents, for each link in our sample, the number of *normal* and *rapid* freight trains travelling in both directions on the link, as planned by the tactical model for the initial case. It also displays the train flows estimated by our method from the product flows determined by the same transportaion plan, and the estimated link capacity. Table 13 displays the corresponding over-the-line delays computed (in minutes), from the train flows of Table 12, by the tactical model, the Petersen's delay formula and our own approximated polynomial delay.

Link Number	Normal Freight	Rapid Freight	Total Planned	Estimated Capacity	Estimated T_a
1	39	21	60	213	54
2	66	30	96	194	93
3	14	0	14	73	8
4	98	0	98	124	98
5	75	0	75	48	78
6	6	0	6	224	6
7	10	0	10	261	12
8	21	0	21	147	20
9	10	0	10	234	8
10	12	0	12	25	6

Table 12 Total Train Link Flows

Several observations may be made based on the figures displayed in tables 12 and 13. First, the results reconfirm that the polynomial delay offers a very good approximation of the delay formula proposed by Petersen. One also notes that, while Petersen's formula computes a negative delay, the delay function that we propose manages successfully the case where a train flow superior to the link capacity has been assigned to a particular link (e.g. link 5).

When comparing the polynomial and the tactical planning delay formulae, one should first note that the capacities used in our experiments were estimated by using approximate figures only. Since the capacity information was not required in the tactical model (it is implicitly defined into the coefficients of the over-the-line delay function) and, thus, was not readily available, we had to make some educated guesses. These are clearly wrong for some of the links (links 4 and 5, particularly), which biases the results for these links, at least. Yet, the overall performance of the polynomial approximation function is very satisfactory indeed.

Link Number	Planned Rapid	Planned Normal	Polynomial Delay	Petersen's Formula
1	339,42	415,62	371,71	391,83
2	21,96	—26,94	30,78	35,78
3	-	560,40	460,11	458,87
4	-	806,94	3140,67	2669.74
5	-	702,66	133541,27	-779,34
6	-	72,90	53,39	53,28
7	-	100,98	76,18	75,55
8	-	226,98	187,23	184,17
9	-	235,68	174,54	174,78
10	-	73,44	66.33	68.77

Table 13 Planned and Estimated Over-the-Line Train Delays

The approximation formula generally underestimates the delay computed by the tactical planning function and relative errors between 10% and 27% are observed between the delays computed by the two functions. This is very satisfactory when one considers the cascading approximations implicit in the polynomial over-the-line delay: the local train approximation and the polynomial approximation of a delay function that, in turn, approximates the function used in the tactical model. Thus, an improvement in any of these approximations, especially a better fit of the train link flow estimation and a more precise computation of the link capacity figures, should greatly improve the overall precision of the over-the-line delay estimation.

It is also interesting to note that, as expected, the approximation performs better when the line is not too congested, since the polynomial delay "exagerates" the behaviour of the congested part of the delay curve. In this case, the presence on the network of several train types with different speeds and priorities has no significant impact on the degree of fitness of the polynomial delay function. This is a very encouraging result and augures well for further studies of the approximation of over-the-line delays in high congestion situations and when several train types are defined on the network.

7.5 CONCLUSIONS

We have presented and analysed approximation formulae for the total train flow and for the over-the-line train delay for the links of a rail network that may be used in a model for the strategic planning of a national or regional multimode multicommodity freight transportation system.

The approximations have proved to be accurate and, thus, totally adequate for strategic planning modeling purposes. We have, therefore, also proposed possible utilization modes for the various formulae and their corresponding calibration procedures.

We noted that the precision of the approximation formulae is quite dependent on the accuracy of the input data and we suggest, therefore, that particular attention should, at least, be paid to:

- the estimation of the link capacity;
- the estimation of the *by product train weights* ;
- the utilization of the link correcting factor to adequately adjust the train approximation formula;
- the utilization of the most precise formula possible as starting point for the calibration of the over-the-line polynomial delay function.

This will improve the overall quality of the approximations and thus the validity of the flows simulated by the global network planning model.

7.6 BIBLIOGRAPHY

Crainic, T.G. (1982), *Un modèle de planification tactique pour le transport ferroviaire des marchandises* , Ph.D. thesis, publication #247, Centre de recherche sur les transports, Université de Montréal.

Crainic, T.G. (1987), *Rail tactical planning: issues, models and tools*, Proceedings of *Freight Transport Planning and Logistics*, Bressanone Italy.

Crainic, T.G., Ferland, J.A. and Rousseau, J.M. (1984), *A tactical planning model for rail freight transportation* , **Transportation Science** 18 (2), 165-184.

Crainic, T.G., and Querin, L. (1987), *Approximation of Over-the-Line Delay for Single-Line Single-Speed Freight Rail Transportation*, publication #519, Centre de recherche sur les transport, Université de Montréal.

Petersen, E.R. (1974), *Over-the-road transit time for a single track railway* , **Transportation Science** 8, 65-76.

Roy, M.-C. (1986), *Modélisation stratégique du transport des marchandises au Brésil: le mode ferroviaire et l'analyse de matrices de demande*, M. Sc. thesis, publication #485, Centre de recherche sur les transports, Université de Montréal.

7.7 APPENDIX - PROCEDURES

In this appendix, we present four procedures that may be used, either individually for evaluation purposes, or as part of the calibration procedure described in Section 7.3 to generate polynomial delay functions that approximate delay formulas (here, Petersen's (1974)) that may be more precise but less computationally or analytically suitable.

There is a procedure for each of the following operations, required by the steps of the calibration procedure:

1. Generate a set of points (total link train flows) and compute link delays;
2. Evaluate the parameters of a polynomial delay function of specified degree and compute the corresponding link delays;
3. Compare two delay fonctions;
4. Evaluate link delays using a specified polynomial function.

The programs are written in FORTRAN. The procedures treat all the links of specified link sets.

INITIAL DATA FORMAT

The file LINKS contains the initial information associated with the links for which delays (polynomial or not) have to be evaluated. There is a record for each link, in the following format (all file descriptions in this Appendix display the *name* of the field in the first column, the *format* in the second column and the *description* of the field in the third):

NO1	(I4)	Origin node;
NO2	(I4)	Destination node;
FRT	(F10.5)	Free running time (in minutes);
CAP	(F10.5)	Capacity (weekly);
t_a	(F12.5)	Train flow from NO1 to NO2;
t_b	(F12.5)	Train flow from NO2 to NO1;
DIST	(I5)	Link length;
NTP	(I4)	Number of passenger trains on the link.

✓

STEP 1 - GENERATE DELAYS

This procedure, called FPETER, computes, for a set of links, the over-the-line delays for various train flow values using the Petersen delay formula.

The procedure considers all integer values for the total train flow, $T_a = t_a + t_{\bar{a}}$, between 0 and 92% of each link's capacity. For *even* total train flow values, only one delay evaluation is performed, supposing equal train flows in each direction. For *odd* values, two delays are computed: one using $t_a = (T_a + 1)/2$ and $t_{\bar{a}} = (T_a - 1)/2$, and another using the reverse values. The average of these two delays is later used by the comparing procedures of Step 3.

The input of the procedure is the file LINKS. The output of the procedure is the file PETER which contains, for each link, a number of records specifying the delay values for all considered train flows. The format of the file is as follows:

- Header record, description of link **a**

NO1	(I4)	Origin node;
NO2	(I4)	Destination node;
FRT	(F10.5)	Free running time (in minutes);
CAP	(F10.5)	Capacity (weekly);

- Empty record

- One point per record

T_a	(I15)	Total train flow;
D_a	(F15.5)	The computed delay.

- Last record for the link

0 0 (2I15)

The file PETER is used as input by the two next procedures. It may also be used to draw graphs of delay versus total train flow for some or all links in the file. These graphs, in turn, may then be used to choose values for the exponent of the estimated polynomial delays.

It is noteworthy that, as long as the format of the input and output files are respected, another procedure may be substituted to FPETER if one wants to try a different delay formula. Yard delays could thus be approximated if the appropriate procedure is used instead of FPETER. It is of course understood that the new formula gives a delay of the same general shape (see Figure 1) as the current one.

STEP 2 - EVALUATING PARAMETERS

This procedure, called FPHIS, computes the values of the parameters ϕ_1 and ϕ_2 of the polynomial delay, given a specified polynomial degree **exp** . The computation method has been described in Section 7.3. If desired, the procedure also produces the over-the-line delays generated using this polynomial.

The procedure accepts the file PETER (and the degree of the polynomial) as input. The output of the procedure is the file PHIS which contains the values of the parameters and, eventually, the computed delays.

The contents of the output file is controlled by the value of the logical variable $LONG$. The long form, $LONG = true$, has the following format:

- Header record, description of link **a**

NO1	(I4)	Origin node;
NO2	(I4)	Destination node;
FRT	(F10.5)	Free running time (in minutes);
CAP	(F10.5)	Capacity (weekly);
exp	(F10.5)	Degree of the polynomial delay;
ϕ_1	(F10.5)	Linear parameter of the polynomial;
ϕ_2	(F10.5)	Congestion parameter of the polynomial.

- Empty record
- One point per record:

T_a	(I15)	Total train flow;
D_a	(F15.5)	The delay computed using the estimated parameters.

- Last record for the link

0 0 (2I15)

In the short form, $LONG = false$, only the estimated parameters for each link (in the previous header link record format), plus the average value of ϕ_2, are written:

- Header record link1
- Header record link2
- ...
- Header record last link
- Average value of ϕ_2 (F15.5)

STEP 3 - COMPARING DELAYS

The procedure FECARTS compares a given delay formula with its polynomial approximation. Specifically, it computes, for all links, the average relative error (ARE) and the maximum relative error (MRL) of the polynomial delay as compared to the initial formula. The procedure FECARTS accepts as input the files PETER and PHIS (in the *long* format) and outputs the errors by link and, eventually, by total train traffic value.

The contents of the output file, called ECARTS, is again controlled by a logical variable. The short version, $LONG = false$, gives, for each link, the mean and relative errors, in the following format:

- Header record: see the file PHIS;
- Error record:
  ARL, MRL (2F15.5).

The long version gives, in addition, the relative error for each point:

- Header record, description of link **a**

NO1	(I4)	origin node;
NO2	(I4)	destination node;
FRT	(F10.5)	free running time (in minutes);
CAP	(F10.5)	capacity (weekly);
exp	(F10.5)	degree of the polynomial delay;
ϕ_1	(F10.5)	linear parameter of the polynomial;
ϕ_2	(F10.5)	congestion parameter of the polynomial.

- One point per record:

T_a	(I4)	total train flow;
RE	(F15.5)	relative error for that train flow.

- Last record for the link:
  ARL, MRL (2F15.5).

STEP 4 - EVALUATING POLYNOMIAL DELAYS

The procedure FPOLY computes, for all specified links, the polynomial over-the-line delay.

The procedure accepts the file LINKS and values for **exp** , ϕ_1 and ϕ_2 as input, and outputs, for each link, the delay computed with the specified polynomial.

The output of the procedure is the file POLY, that has exactly the same structure and format as the (long) file PHIS generated by the procedure FPHIS in Step 2. Consequently, the output of FPOLY may be given as input to FECARTS to evaluate the accurateness of any given polynomial.

CHAPTER EIGHT

ON THE CONSIDERATION OF CAPACITY CONSTRAINTS IN THE STAN MULTI-MODE MULTI-PRODUCT ASSIGNMENT PROCEDURE

The route and amplitude of freight flows are often directed and coerced by restrictions imposed by the limited handling capacity of the various infrastructure facilities: rail lines, pipelines, loading/unloading equipment, etc. Furthermore, quite often modes that are most cost or energy efficient also present more stringent capacity restrictions. Railways and navigation are typical examples of this situation.

Planning methods and tools must therefore take into account the capacity restrictions present on the links and modes of the various modal transportation networks, especially when minimum cost flow assignments are contemplated.

For STAN, a choice has been made (see Chapter 1) to consider capacity restrictions in the multi-mode multi-product assignment model, not as explicit constraints, but via terms in the objective function.

When delays are used as an assignment criteria, capacity restrictions may adequately be considered by using appropriately defined volume/delay functions. These functions model the time necessary to pass through a given facility according to its capacity, physical characteristics and the volume of traffic attempting to use it. The passing-through time is sharply increased (toward infinity) when volumes tend toward the capacity, thus effectively enforcing the capacity restriction. This approach, often used for modeling rail transportation, has been examined in Chapters 4 and 7.

This chapter is dedicated to the analysis of the more delicate situation, when capacity restrictions have to be combined with, usually linear, cost functions. We first present a summary of the problem and examine some direct but not to efficient solution methods. We then reformulate the problem and use standard penalty methods to solve it. We then compare these methods in order to select the best one to use in the STAN assignment framework.

8.1 THE NUMERICAL PROBLEM CONSIDERED

A single product is transported from A to B on the following multi-modal network (Figure 1):

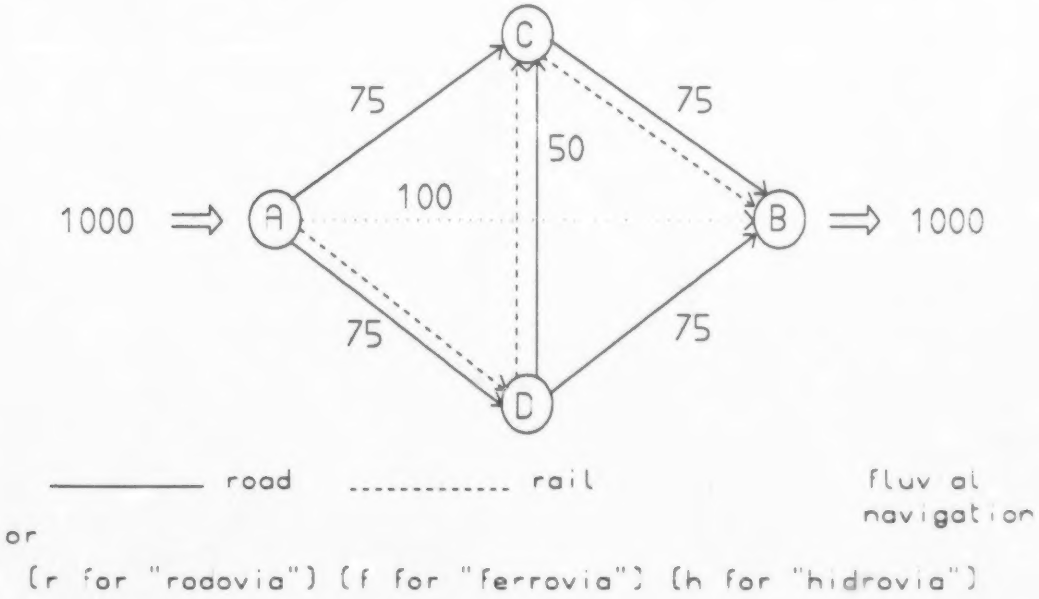

C
75
75
50
1000
A
100
B
1000
75
75
D
road
rail
fluvial navigation
or
(r for "rodovia") (f for "ferrovia") (h for "hidrovia")

Figure 1 The Network of the Problem Considered

The length of the links is given in Figure 1, as well as the demand of the product that must be shipped from A to B. Modal transfers are permitted at nodes C and D between road and rail.

In this example, only the operating costs are considered. These costs are constant on each link and depend only on the length of the link and the mode used. Specifically, they are given by the formulae shown in Table 1.

Mode	Mean cost
r	$30 + 0.03 * \text{length}$
f	$30 + 0.02 * \text{length}$
h	$5 + 0.01 * \text{length}$

Table 1 Original Link Unit Costs

Mode	Path	Total Length
r	A → C → B	150
	A → D → C → B	200
	A → D → B	150
f	A → D → C → B	200
h	A → B	100

Table 2 Access Paths

Relatively high transfer penalties are hindering any change of transportation mode, so there are, in effect, only five ways of shipping the product from A to B, as shown in Table 2.

Without any capacity limitations, the minimal total cost solution is trivially the "all-or-nothing" solution that assigns the demand to the cheapest route, which is through fluvial navigation. The problem we have to consider now is the introduction of capacities for train and fluvial navigation. These capacities are the same on each link and are only mode dependent, as shown in Table 3.

Mode	Capacity
r	none
f	500
h	100

Table 3 Capacity on Links

The capacity constrained problem has a trivial optimal solution : because there is only one available path for each of the capacitated (cheapest) modes, sending 100 units by fluvial navigation, 500 units by rail and the rest by the cheapest road path (either ACB or ADB) yields the total cost optimal assignment. This solution (with ACB chosen as the road path) is summarized in the Table 4.

Since there are no explicit capacity constraints in the STAN minimum cost assignment, these restrictions must by modeled by appropriately chosen unit cost functions. The following functions model the capacity constraints, but are not differentiable nor continuous:

$$S_r(v_r) = 30+0.03*\text{length} \tag{1}$$

$$S_f(v_f) = \begin{cases} 30 + 0.02 * \text{length} & \text{if } v_f \leq 500 \\ M_f & \text{if } v_f > 500 \end{cases} \tag{2}$$

From Node	Through Mode	To Node	Volume	Total Cost
A	r	C	400	6900
A	r	D	0	0
A	f	D	500	5750
A	h	B	100	600
C	r	B	400	6900
C	f	B	500	5750
D	r	C	0	0
D	r	B	0	0
D	f	C	500	5500

Table 4 Optimal Link Flows and Total Cost

$$S_h(v_h) = \begin{cases} 5 + 0.01 * \text{length} & \text{if } v_h \leq 100 \\ M_h & \text{if } v_h > 100 \end{cases} \tag{3}$$

where M_f and M_h are suitably large constants.

The challenge is then to find a continuously differentiable function that fits as closely as possible the shape of the discontinuous "capacity" function shown in Figure 2.

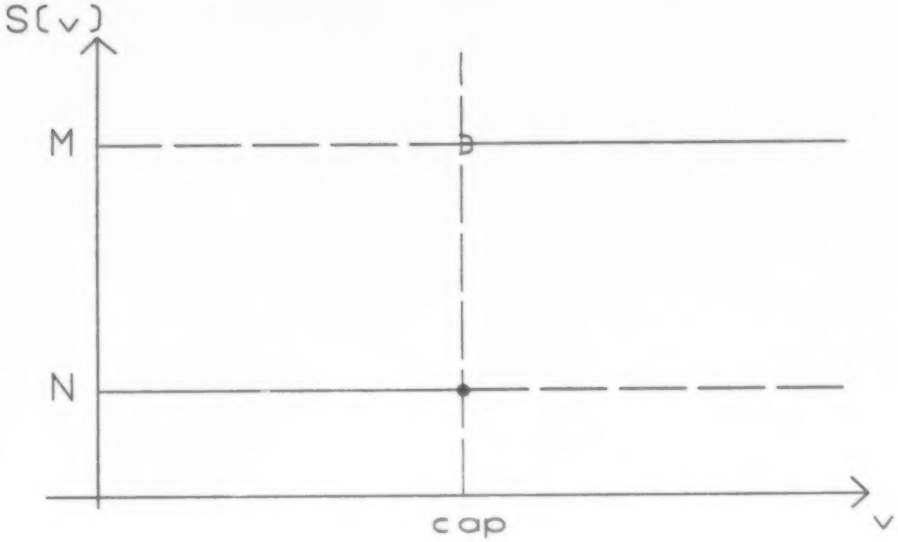


Figure 2 "Capacity" Function

8.2 POSSIBLE APPROXIMATIONS OF THE "CAPACITY" FUNCTION

A convenient function to use is a logit function of the form

$$s(v) = N + \frac{M - N}{1 + e^{-k(v-cap)}} \tag{4}$$

as shown in Figure 3.

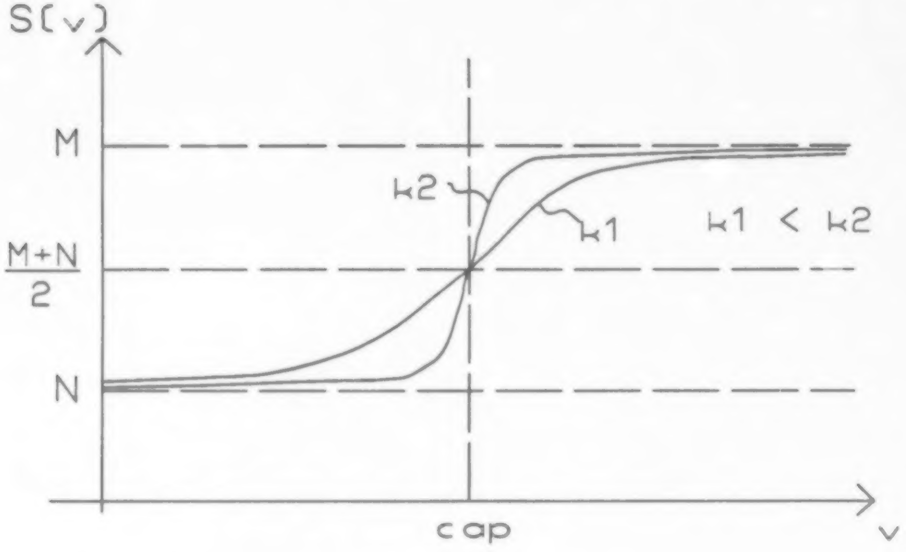


Figure 3 Logit Functions

The major disadvantage of such a function is that it results in a non-convex total cost function. This can cause convergence problems in the assignment algorithm as it is implemented in the STAN system. Indeed, let us inspect the second derivative of (4)

$$s''(v) = \frac{k(M - N)e^{-k(v-cap)}}{1 + e^{-k(v-cap)}}[2 - vk + (2 + vk)e^{-k(v-cap)}] \tag{5}$$

The first term of (5) is always positive, but the second may become negative if v is large enough (in fact as soon as $e^{-k(v-cap)} < 1-4/(vk+2)$, which is true for $v = cap+\epsilon$, $0 < \epsilon << cap$). Figure 4 shows the total cost function when a logit unit cost function of the form (4) is used ($M = 5N = 5*(30+0.02*\text{length})$; length $= 200$; $cap = 500$).

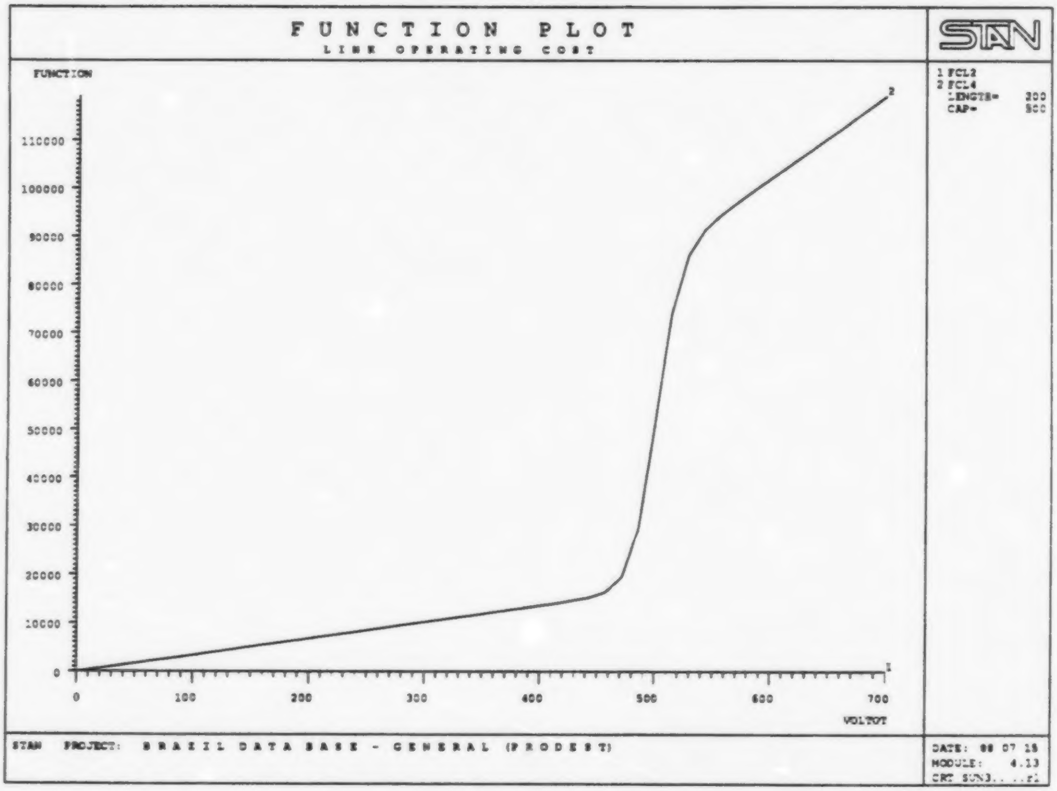


Figure 4 Non-Convex Total Cost

An alternative approach is to approximate the "capacity" function of Figure 2 with a standard polynomial volume-delay function:

$$s(v) = N+\alpha v+\beta(\frac{v}{cap})^{\delta} \tag{6}$$

The linear part should be very flat, so the parameter α was fixed to 0.001. To avoid overflow problems, the value 5 has been given to the exponent δ. The last parameter β has been determined by the equation $\delta(1.2 * cap) = M$ (see Figure 5), which gives $\beta = (M - N - 0.0012 * cap)/(1.2)^5$. With the arbitrary choice $M = 5N$, the link unit cost functions displayed in Table 5 are obtained.

Mode	Unit Cost
r	$30 + 0.03\text{length}$
f	$30 + 0.02\text{length} + 0.001v_f + (47.98 + 0.032\text{length})(v_f/500)^5$
h	$5 + 0.01\text{length} + 0.001v_h + (\ 8.00 + 0.016\text{length})(v_h/100)^5$

Table 5 Polynomial Unit Costs

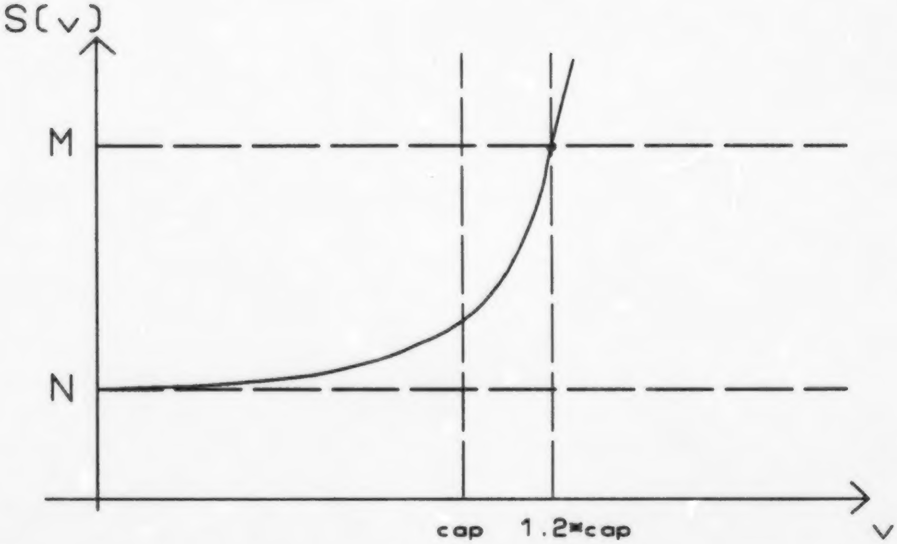


Figure 5 Polynomial Fit

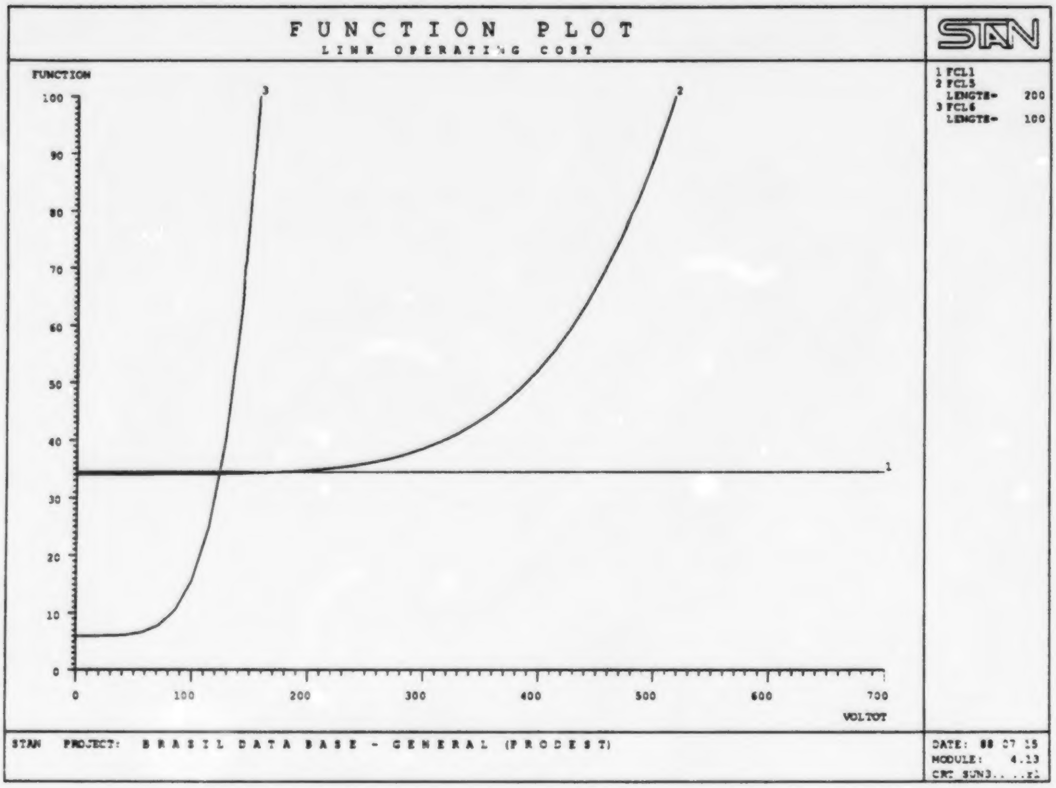


Figure 6 Polynomial Unit Costs

Figure 6 shows the three path unit cost functions (for road, path A → C → B was chosen). A relevant remark is that the unit costs on rail and road are quite similar for volumes not greater than 220 (in fact, the unit cost for the road path is equal to 34.5 and the unit cost under capacity for rail is 34.0 ! ...).

The corresponding marginal cost functions are shown in Figure 7. One can approximatively find the optimal volumes by equalizing marginal costs for each mode,

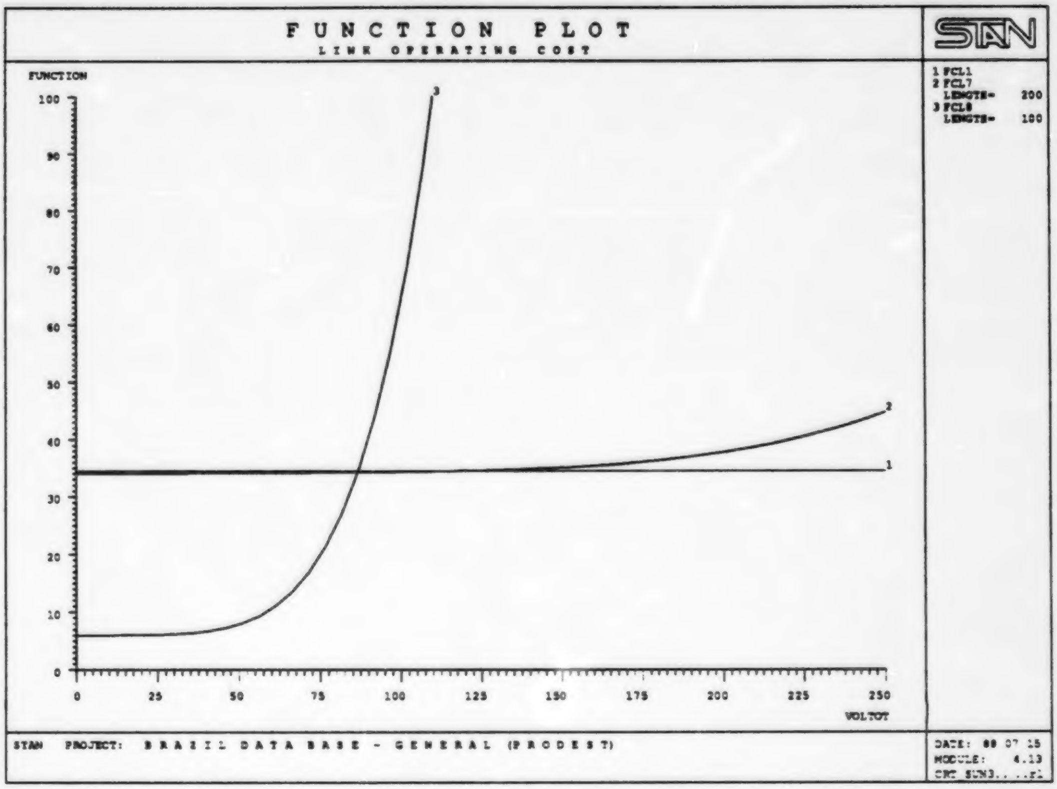


Figure 7 Polynomial Marginal Costs

which gives:

$$\begin{cases} v_f^* \cong 77.0 \\ v_h^* \cong 87.5 \\ v_r^* \cong 1000 - (77.0 + 87.5) = 835.5 \end{cases} \tag{7}$$

These volumes are very far from the true optimal solution (see Table 4). The STAN assignment algorithm found the solution ($v_r = 831.71, v_f = 99.66, v_h = 86.63$) after ten iterations and ($v_r = 817.96, v_f = 95.29, v_h = 86.75$) after one hundred iterations.

A conclusion to be drawn from that example is that the polynomial functions increase too fast for flows less than the capacity. Hence, it suggests the use of the following function that fits better the flat part of the "capacity" function (Figure 2) for values of flow less than capacity:

$$S(v) = \begin{cases} a & \text{if} \quad \frac{v}{cap} \leq 0.8 \\ b + c(\frac{v - 0.8cap}{cap - 0.8cap})^d & \text{if } 0.8 < \frac{v}{cap} \leq 1.2 \\ e + f(\frac{v}{cap}) & \text{if } 1.2 < \frac{v}{cap} \end{cases} \tag{8}$$

We call these functions "mixed-polynomials". See Figure 8.

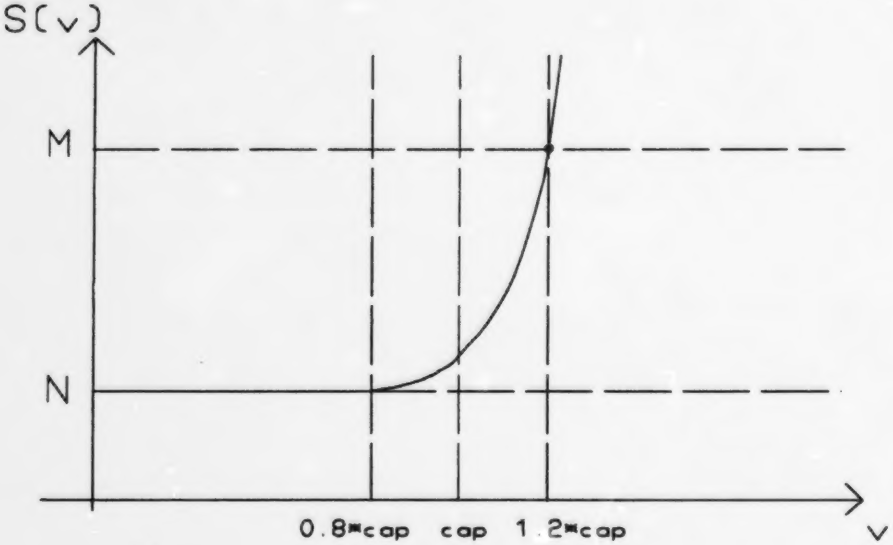


Figure 8 Mixed Polynomial Fit

From the relations :

$$S\ (0) = N \tag{9}$$

$$\lim_{v \nearrow 1.2cap} S\ (v) = M \tag{10}$$

$$\lim_{v \searrow 1.2cap} S\ (v) = M \tag{11}$$

$$\lim_{v \nearrow 1.2cap} S'(v) = \lim_{v \searrow 1.2cap} S'(v) \tag{12}$$

it is possible to determine the different parameters of (8):

$$a = b = N \tag{13}$$

$$c = \frac{M - N}{2^d} \tag{14}$$

$$e = M - 3d\,(M - N) \tag{15}$$

$$f = 2.5d\,(M - N) \tag{16}$$

By setting d to 5 and M to $5N$, we obtain the unit cost:

$$s(v) = \begin{cases} N & \text{if} \quad \frac{v}{cap} \le 0.8 \\ N(1 + 0.125(5\frac{v}{cap} - 4)^5) & \text{if } 0.8 < \frac{v}{cap} \le 1.2 \\ N(-55 + 50\frac{v}{cap}) & \text{if } 1.2 < \frac{v}{cap} \end{cases} \tag{17}$$

Figures 9 and 10 show the corresponding mean and marginal path unit costs. Again, the optimal volumes may be easily computed by equalizing the marginal costs of the paths in Figure 10 :

$$\begin{cases} v_f^* \cong 425 \\ v_h^* \cong 100 \\ v_r^* \cong 1000 - (425 + 100) = 475 \end{cases} \tag{18}$$

This solution is much more satisfactory than (7). The flows obtained with the STAN assignment procedure after 10 and 100 iterations are respectively:

$$(v_r = 634.51, v_f = 264.31, v_h = 101.18)$$

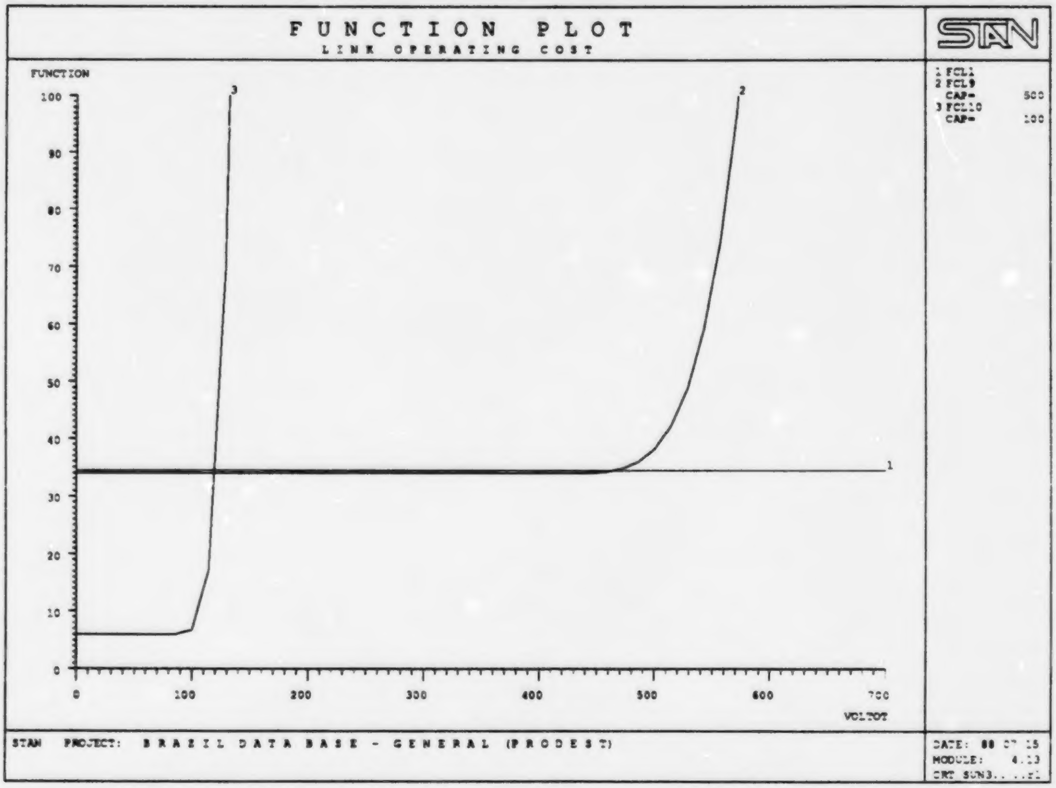


Figure 9 Mixed Polynomial Mean Costs

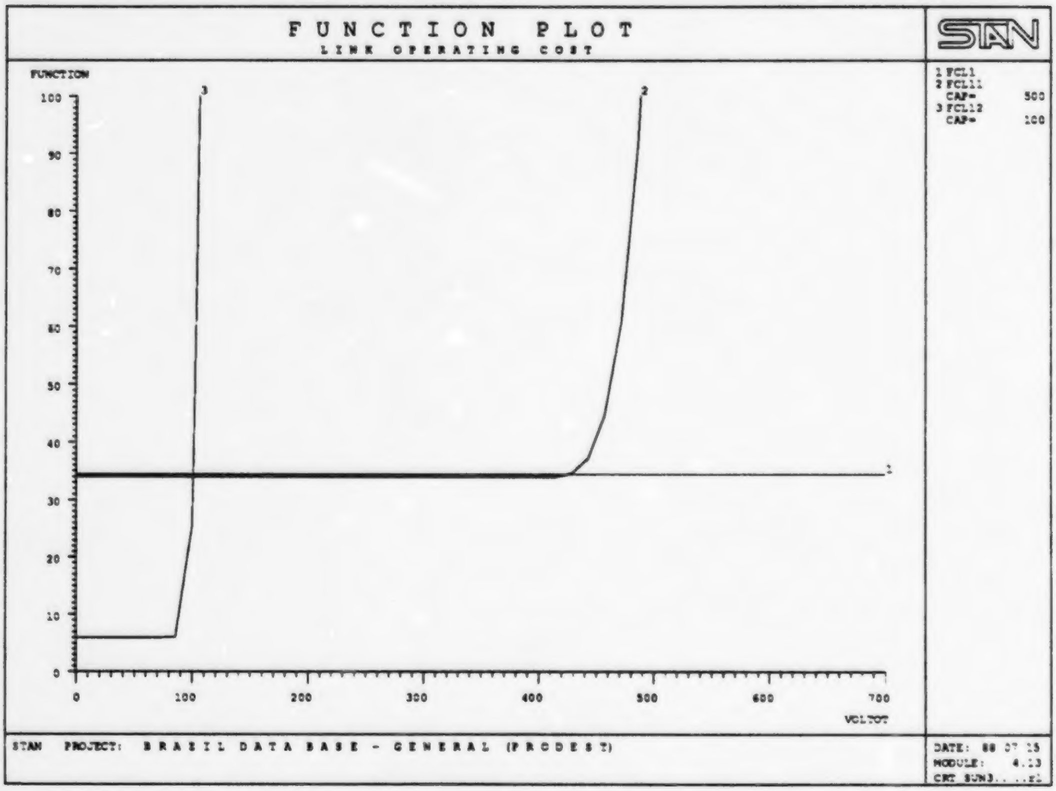


Figure 10 Mixed Polynomial Marginal Costs

$$(v_r = 525.96, v_f = 372.36, v_h = 101.68).$$

These results indicate that there may be a convergence problem with the linear approximation algorithm. We discuss this phenomenon in more detail in the next section. However, if the total costs (Table 6), are examined, the solution is quite good. This is an indication that the problem is *ill-conditioned* due to the fact that the two path unit cost for road and rail are nearly equal.

Solution	Total cost	Relative error (%)
10 iterations	31585	0.22
100 iterations	31530	0.05
near optimal	31514	-

Table 6 Total Cost (Mixed Polynomial)

8.3 OTHER SOLUTION APPROACHES

In this section, we reformulate the problem in a different way, which is equivalent to the problem considered so far, but is much simpler to analyze.

As was noted before (see Table 2), there is only one path available for rail and fluvial navigation between A and B. Since the road unit cost is constant and since there is no capacity on road links, only the shortest path has to be considered for this mode, that is either A → C → B or A → D → B. Path A → C → B is chosen arbitrarily. The problem is then equivalent to shipping 1000 units of product between A and B on the network shown in Figure 11:

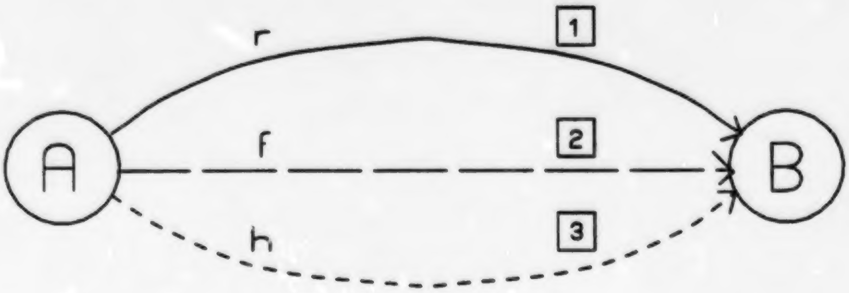


Figure 11 Equivalent Network

The attributes of the links are given in the Table 7:

Link no	Mode	Length	Capacity	Fixed Unit Cost
1	r	150	none	$k_r = 34.5$
2	f	200	500	$k_f = 34$
3	h	100	100	$k_h = 6$

Table 7 Equivalent Network Attributes

The minimal total cost assignment can be expressed now as a linear optimization problem:

$$Min \quad F = k_r v_r + k_f v_f + k_h v_h \tag{19}$$

$$s.t. \quad v_r + v_f + v_h = 1000 \tag{20}$$

$$v_f \leq 500 \tag{21}$$

$$v_h \leq 100 \tag{22}$$

$$v_r, v_f, v_h \geq 0 \tag{23}$$

with the optimal solution :

$$(v_r^* = 400, v_f^* = 500, v_h^* = 100); \quad F^* = 31400.$$

Equations (20) and (23) are the usual conservation of flow and nonnegativity contraints. Problem (19)-(23) is in standard form for the STAN assignment procedure except for the two capacity constraints (21) and (22). To get rid of these two constraints, they may be introduced in the objective function as a supplementary penalty term.

Consider the new optimization problem :

$$Min \quad G_\mu = F + \mu\{(\max(0, v_f - 500))^2 + (\max(0, v_h - 100))^2\} \tag{24}$$

s.t. (20) and (23), where μ is an appropriate penalty parameter.

The penalty method (Luenberger, 1984) consists of solving (24) for increasing penalty factors μ_k, where k is an iteration index, yielding a sequence $(v^k)_k$ for which any limit point is a solution of (19)-(23).

(24) is directly usable in the STAN assignment algorithm on the condition that G_μ can be expressed as a total cost of the form $\sum_{modes\ m} s_m(v)v_m$. The objective function (24) may be rewritten as:

$$G_\mu = k_r v_r + \{k_f + \mu\frac{(\max(0, v_f - 500))^2}{v_f}\}v_f + \{k_h + \mu\frac{(\max(0, v_h - 100))^2}{v_h}\}v_h$$

The unit cost functions to use for specifying problem (24) in STAN format are then:

$$s(v) = \begin{cases} k & \text{if } v \le cap \\ k + \mu(v - 2cap + \frac{cap^2}{v}) & \text{if } v > cap \end{cases} \tag{25}$$

as shown in Figure 12.

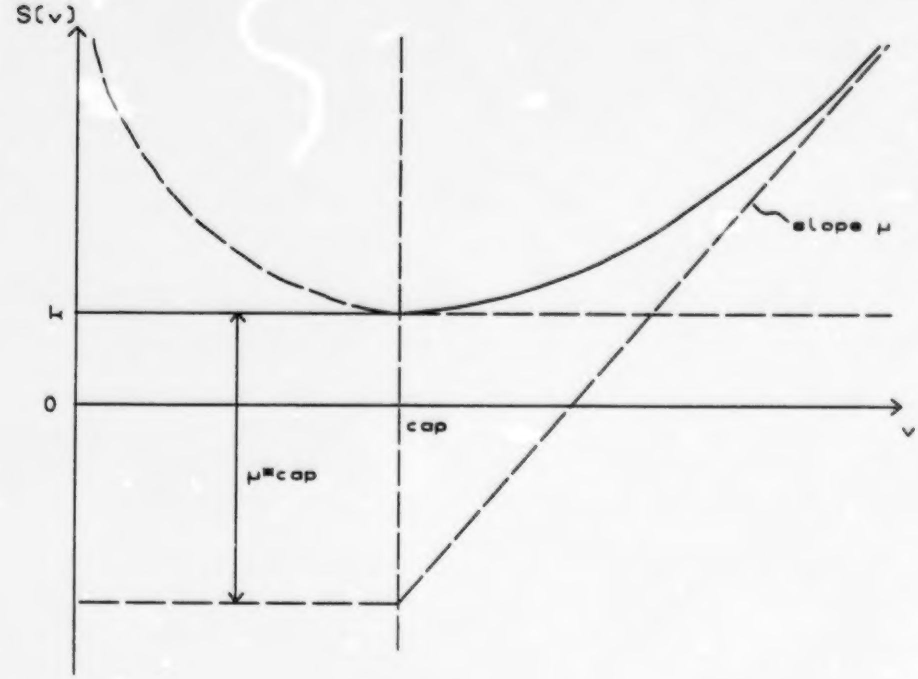


Figure 12 Unit Cost Penalty Function

The corresponding marginal cost, $c(v) = s(v) + s'(v)\ v$, is easily derived from (25):

$$c(v) = \begin{cases} k & \text{if } v \le cap \\ k + 2\mu(v - cap) & \text{if } v > cap \end{cases} \tag{26}$$

The optimal solution of (24) is again obtained by equalizing the path marginal costs:

$$c_r^* = c_f^* = c_h^*$$

or :

$$34.5 = 34 + 2\mu(v_f - 500) = 6 + 2\mu(v_h - 100)$$

which yields

$$v_f^* = 500 + \frac{34.5 - 34}{2\mu} \tag{27}$$

$$v_h^* = 100 + \frac{34.5 - 6}{2\mu} \tag{28}$$

$$v_r^* = 1000 - (v_f^* + v_h^*) \tag{29}$$

We experimented with two values for the penalty parameter: $\mu_1 = 1$ and $\mu_2 = 10$. Figures 13 to 16 show the different mean and marginal costs.

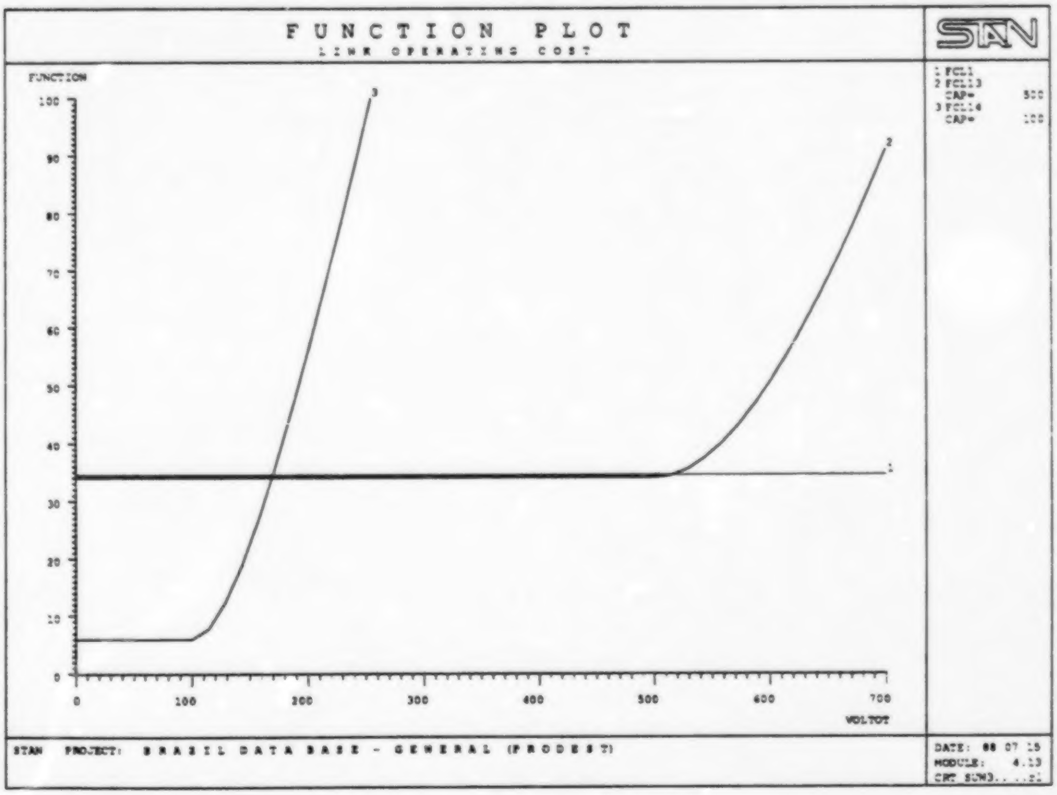


Figure 13 Unit Costs ($\mu = 1$)

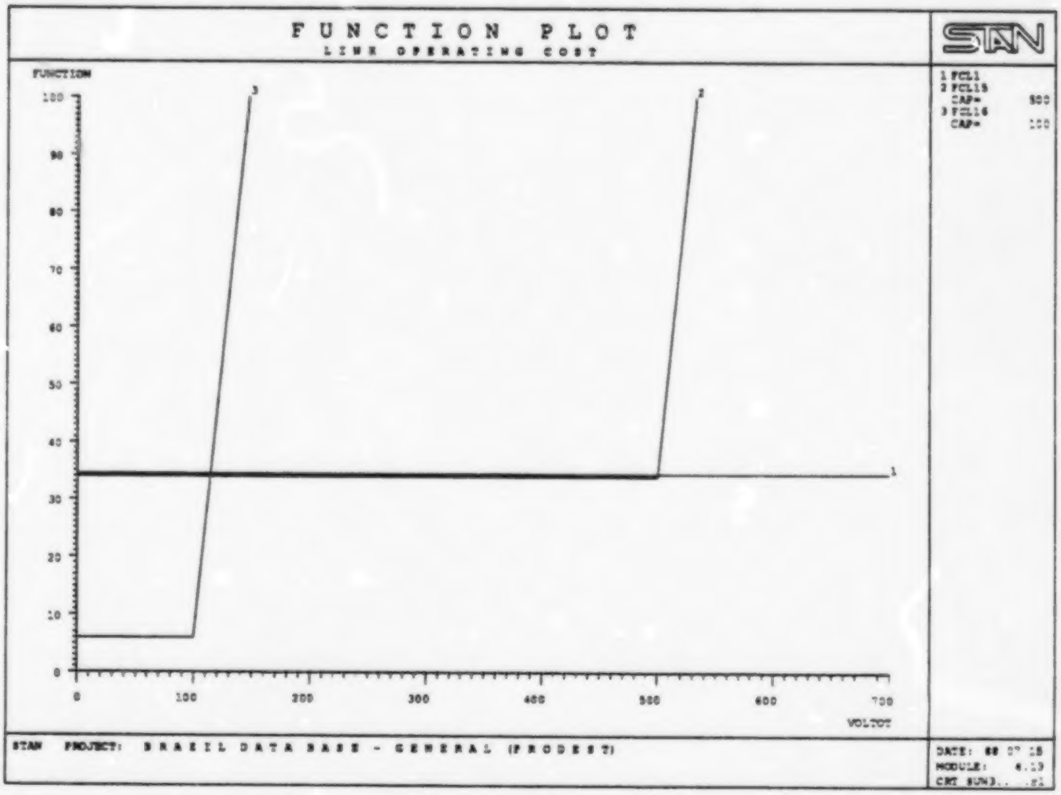


Figure 14 Marginal Costs ($\mu = 1$)

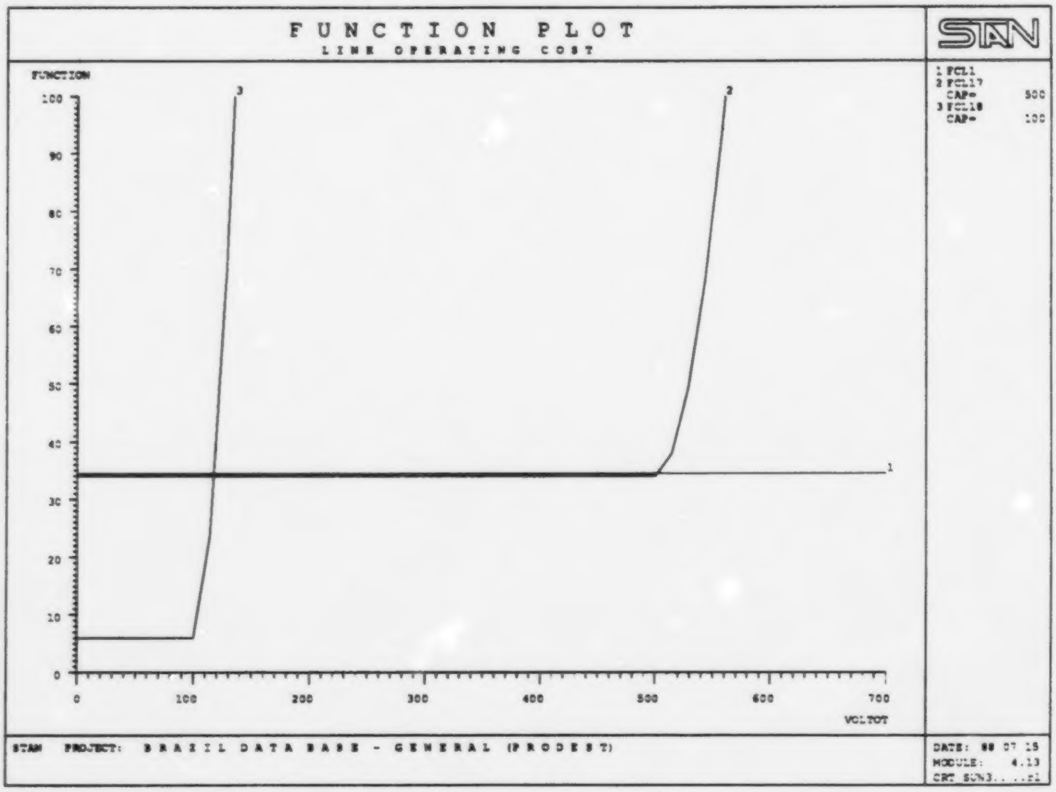


Figure 15 Unit Costs ($\mu = 10$)

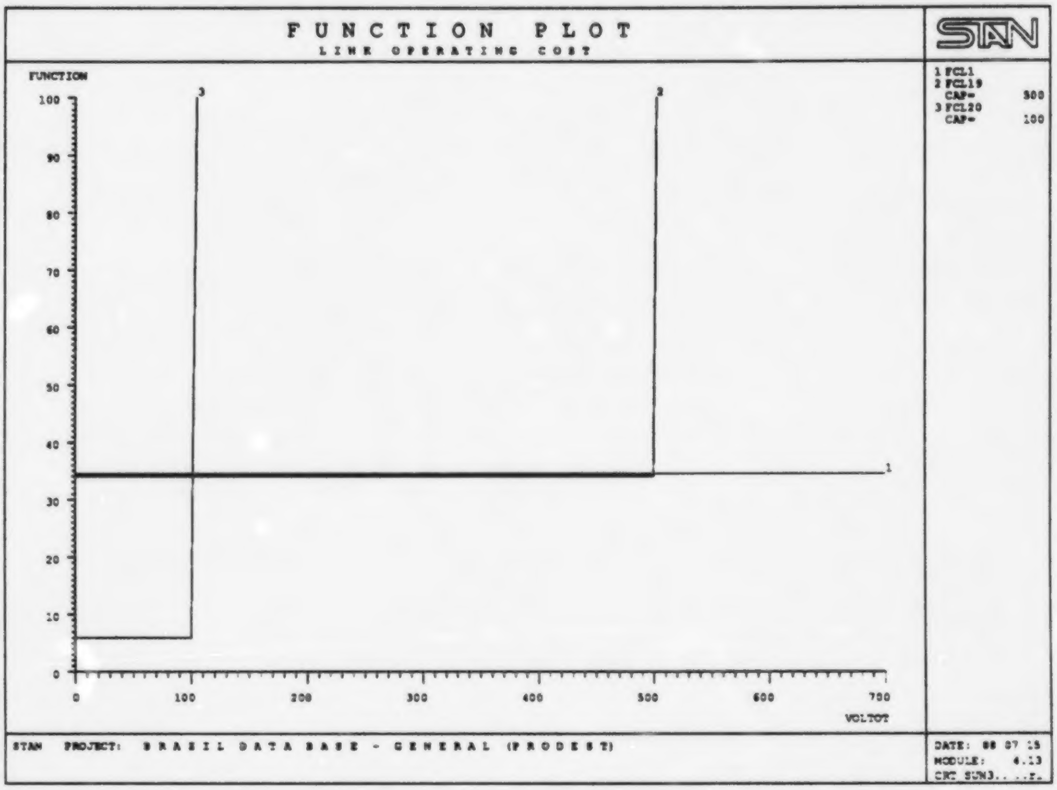


Figure 16 Marginal Costs ($\mu = 10$)

Table 8 (I and II) summarizes the results we obtained by using the assignment algorithm in STAN.

It is well known that the solutions of (24) are not necessarily primal feasible. Since there are only capacity penalty terms in (24), we could reformulate that problem, by reducing the capacities, to get a new one that would be closer to the original problem (19)-(23). Considering (27)-(29), the following formula was tried for the modified capacity:

$$CAP' = (1 - \frac{0.2}{\mu} * \frac{SU_{max} - SU_l}{SU_{max}})CAP \tag{30}$$

where: SU_l = fixed unit cost on link l by unit of length

SU_{max} = maximum fixed unit cost by unit of length

The following relations hold :

$$(1 - \frac{0.2}{\mu})CAP \leq CAP' \leq CAP \tag{31}$$

The results that were obtained with the modified capacities can also be found in Table 8 (III, IV). The optimal solutions of the modified problem (24) are primal feasible now.

	μ	Capacity	Iteration	Volumes			Total Cost	Rel. Error
				r	f	h		(%)
I	1	CAP	10	462.83	423.04	114.13	31235	0.122
			100	387.15	498.60	114.25	31198	0.003
			optimal	385.50	500.25	114.25	31197	-
II	10	CAP	10	461.00	437.68	101.31	31411	0.099
			100	444.29	454.31	101.40	31402	0.070
			optimal	398.55	500.03	101.42	31380	-
III	1	CAP'	10	427.64	473.49	98.87	31638	0.003
			100	427.05	473.70	99.25	31638	0.003
			optimal	425.50	475.25	99.25	31637	-
IV	10	CAP'	10	458.44	442.25	99.31	31466	0.079
			100	441.62	458.98	99.40	31457	0.051
			optimal	402.52	497.53	99.95	31441	-

Table 8 Penalty Results

In each of the trials with the penalty functions, the same convergence problem, experienced with the polynomial fit, reocurred: the volume on the railroad is underestimated even after a hundred iterations. We attempt now to explain this phenomenon. In order to do so, we use the problem (24) with $\mu = 10$. Table 9 summarizes the first three iterations of the STAN algorithm. As it is easy to see, a quasioptimal solution (in terms of total cost) is found after only two iterations. However, due to the fact that the road and rail path unit cost are very similar, the corresponding link volumes are far from their optimal value, the rail flow being too small. The next descent direction should normally correct the low flow on the rail link, but the d_h component is non-zero and the step in this direction is restricted to very small values; the flow $v_h = 101.5$ corresponds to the steep part of the fluvial navigation marginal cost function (see Figure 16).

The results of Table 8, indicate that the convergence to flow optimality is better if μ is small , as the slope of the marginal cost function is smaller (see Figure 14). In concluding, it may be said that the penalty factor should be chosen large enough to fit the original problem as precisely as possible and not too high to prevent convergence problems. The solution of choosing μ relatively small and reducing capacity seems to be a good compromise in order to obtain reasonable solutions for ill-conditioned problems.

8.4 COMPARISONS

The comparisons between the different solution approaches that were tried are summarized in the following tables. Table 10 gives the best approximative solution. Table 11 indicates the rate of convergence and finally Table 12 shows how accurate the approximate problem is.

Iteration	Volumes			Mean costs			Marginal costs			Total cost
Number	v_r	v_f	v_h	S_r	S_f	S_h	C_r	C_f	C_h	$\sum_i S_i v_i$
-	0	0	0	34.5	34	6	34.5	34	6	-
0	0	0	1000	34.5	34	8106	34.5	34	18006	8106000
1	0	699.3	300.7	34.5	602	1346	34.5	4020	4020	825590
2	662.5	236.5	101.5	34.5	34	6.22	34.5	34	36.0	31512
3	660.8	238.0	101.2	34.5	34	6.15	34.5	34	30.8	31511
optimal	398.55	500.03	101.42	34.5	34	6.20	34.5	34.5	34.5	31380

Iteration	Descent direction			Step	Relative error
Number	d_r	d_f	d_h	λ	%
-	-	-	-	1	-
0	0	1000	-1000	0.6993	25731.74
1	1000	-699.3	-300.7	0.6625	2530.94
2	-662.5	764	-101.5	0.0026	0.42
3	-	-	-	-	-
optimal	-	-	-	-	-

Table 9 First Iterations with $\mu = 10$ Penalty Function

Method	Maximum absolute relative error on true optimal link flows (%)	Mean relative error on true optimal link flows (%)	Absolute relative error on true optimal total cost (%)	Absolute relative error on optimal total cost (%)
Polynomial	103.4	65.6	1.851	0.006
Mixed poly.	58.6	35.6	0.268	0.220
Penalty $\mu = 1$	15.7	15.1	1.159	0.122
Penalty $\mu = 10$	15.2	9.7	0.022	0.099
Penalty $\mu = 1$[1]	6.9	4.4	0.143	0.003
Penalty $\mu = 10$[1]	14.6	8.9	0.156	0.079

Table 10 Solution Obtained After 10 Iterations.

Method	Maximum absolute relative error on true optimal link flows (%)	Mean relative error on true optimal link flows (%)	Absolute relative error on true optimal total cost (%)	Absolute relative error on optimal total cost (%)
Polynomial	104.5	66.2	1.847	0.003
Mixed poly.	31.5	19.6	0.051	0.005
Penalty $\mu = 1$	14.3	5.9	1.290	0.003
Penalty $\mu = 10$	11.1	7.2	0.054	0.070
Penalty $\mu = 1$[1]	6.8	4.3	0.111	0.003
Penalty $\mu = 10$[1]	10.4	6.4	0.121	0.051

Table 11 Solution Obtained After 100 Iterations.

---

[1] lower capacity

As was noticed in the previous section, the penalty function with $\mu = 1$ and modified capacity gives the best approximative solution (see Table 10), although the approximate problem is not as good as those with a greater penalty parameter μ (see Table 12). Convergence to optimality is also better with the penalty parameter $\mu = 1$ (see Table 11).

Method	Maximum absolute relative error on true optimal link flows (%)	Mean relative error on true optimal link flows (%)	Absolute relative error on true optimal total cost (%)
Polynomial	108.8	68.7	1.808
Mixed poly.	6.2	3.7	0.040
Penalty $\mu = 1$	6.0	5.9	1.293
Penalty $\mu = 10$	1.4	0.6	0.127
Penalty $\mu = 1$[1]	6.4	4.0	0.108
Penalty $\mu = 10$[1]	0.6	0.4	0.009

Table 12 At Optimality

Clearly the two polynomial approaches (and specially the first one) need a more refined calibration.

---

[1] lower capacity

8.5 CONCLUSIONS

Among the methods of approximating the noncontinuous "capacity" function, the penalty version (24) with penalty factor μ not too high ($\cong 1$) gives relatively satisfactory results. Formula (30) used to get (quasi-)feasible solutions extends easily to a real network. Other fitting functions should be investigated to reach more definitive conclusions.

Due to the ill-conditioned fixed unit cost, some convergence problems arose when solving this problem with the STAN assignment algorithm. However, these difficulties are alleviated as soon as the unit cost on road and rail are of different magnitude. Nevertheless, equal path unit costs may happen in real networks; an acceleration procedure should then be developed in the proximity of the optimal solution to speed up convergence.

8.6 BIBLIOGRAPHY

Luenberger, D.G. (1984), *Introduction to linear and nonlinear programming*, Addison-Wesley.

CHAPTER NINE

A NUMERICAL APPROXIMATION PROCEDURE TO COMPUTE PARTIAL DERIVATIVES OF COST FUNCTIONS

The analytical forms of the arc cost functions, in particular those for the rail mode, may be rather complex expressions. The analytical form of the derivatives are equally complex and rather tedious to specify manually. The evaluation of a derivative from its analytical form may be as demanding as the evaluation of the function itself. A convenient and "user friendly" way, to overcome the analytic specification of the derivatives and partial derivatives, is the employment of a numerical approximation of the function values in order to obtain the derivatives.

9.1 GRAM POLYNOMIALS

Among the many methods that may be used to compute the derivative of a function $f(x)$ at a given point x_0, we have chosen to use a polynomial approximation based on the least squares method which provides, in general, a better fit than that obtained by an interpolation formula (see Kellison (1975), pp.150-151). In order to obtain a least squares fit, the use of orthogonal polynomials is suggested (see Hildebrand (1974), p.348). Among this class of polynomials, the *Gram polynomials* have the following analytical form:

$$P_{m,n}(x) = \sum_{k=0}^{m} \frac{(-1)^{k+m}(m+k)!\,(n+x)!\,(2n-k)!}{(k!)^2(m-k)!\,(n+x-k)!\,(2n)!}$$

where m is the degree of P and $N = 2 \times n + 1$ is the number of points used for function evaluations.

9.2 EQUIDISTANT OBSERVATIONS

The numerical approximation by Gram polynomials may be used when the points on the interpolation abcissa are equidistant. One has, therefore, the following data available:

$$(x_i, y_i) \qquad j = -n, -n+1, \ldots, n$$

with: $$\text{Step} = h = x_{j+1} - x_j \quad \text{(and independent of j)}$$

$$y_j = f(x_i).$$

The least squares approximation may be written as :

$$Q_{m,n}(i) = \sum_{k=0}^{m} \frac{\sum_{j=-n}^{n} y_j P_{k,n}(j)}{\sum_{j=-n}^{n} [P_{k,n}(j)]^2} \; P_{k,n}(i)$$

One has then :

$$f'(x_0) = \tfrac{1}{h} Q'_{m,n}(0) = \tfrac{1}{h} \sum_{i=-n}^{n} \sum_{k=0}^{m} \frac{P_{k,n}(i) P'_{k,n}(0)}{\sum_{j=-n}^{n} [P_{k,n}(j)]^2} \, y_i \qquad (1)$$

These formulae. as well as a code written in FORTRAN. may be refered to in Casaletto et Rice (1978).

9.3 CALIBRATION OF m AND n

Figure 1 depicts the numerical approximation data:

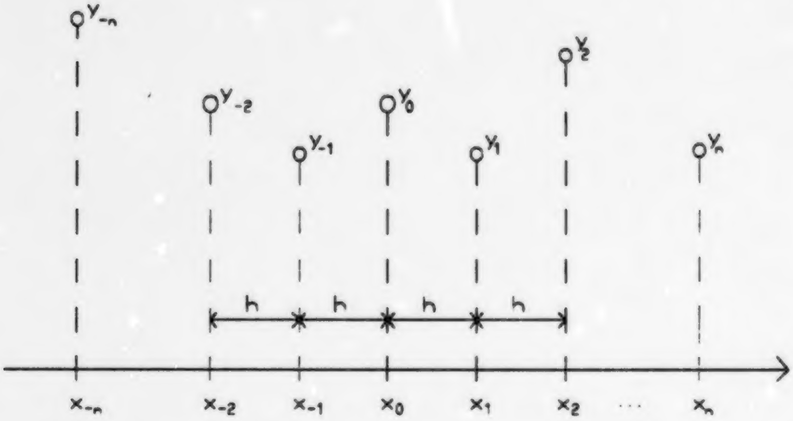

y-n
y-2
y-1
y0
y1
y2
yn
h
h
h
h
x-n
x-2
x-1
x0
x1
x2
...
xn

Figure 1 Equidistant Observation

$N = 2n + 1$ = number of interpolation points

m = degree of the approximation polynomial.

We tested different values of m and n on the following test functions:

1) **EXP** $= e^x$

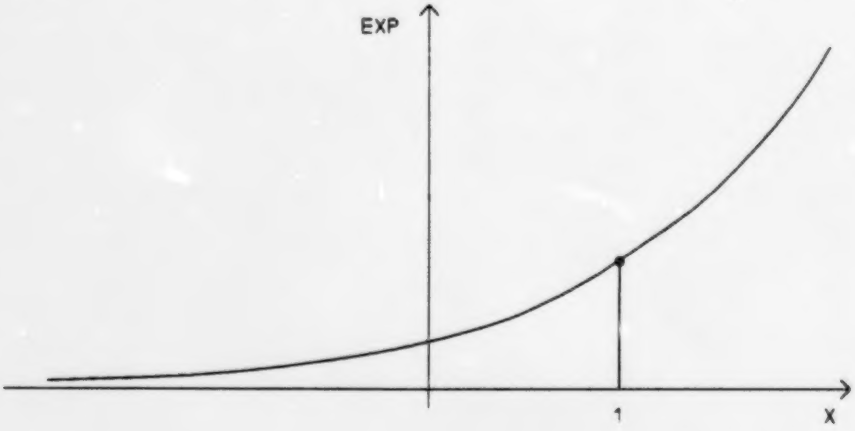


Figure 2 Exponential Test Function

2) SIN = sin(x)

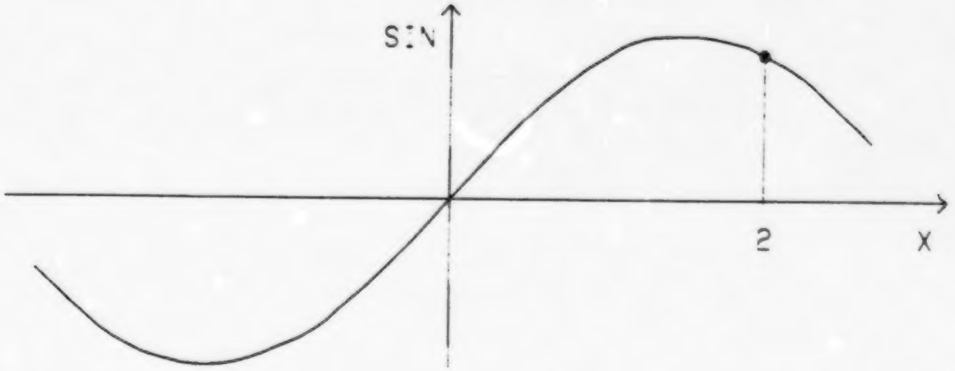


Figure 3 Sinus Test Function

3) TG = tg(x)

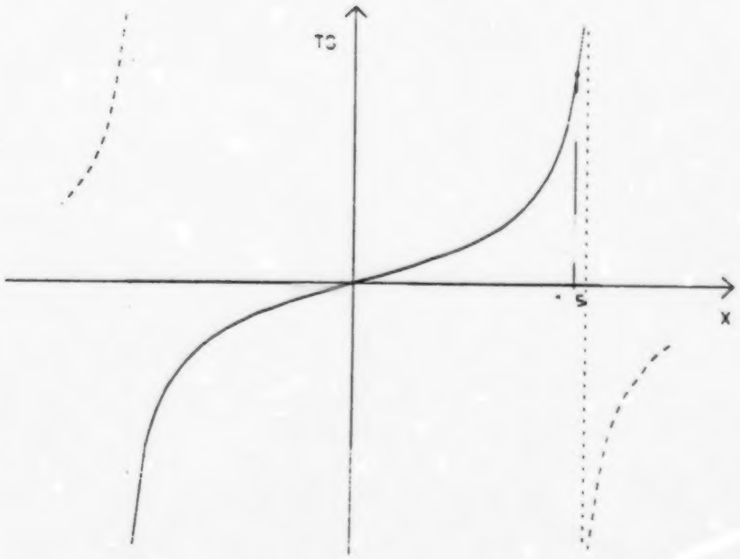


Figure 4 Tangent Test Function

4) VD = volume-delay

$$VD = 2(1 + 5(\frac{x}{500})^{3.8})$$

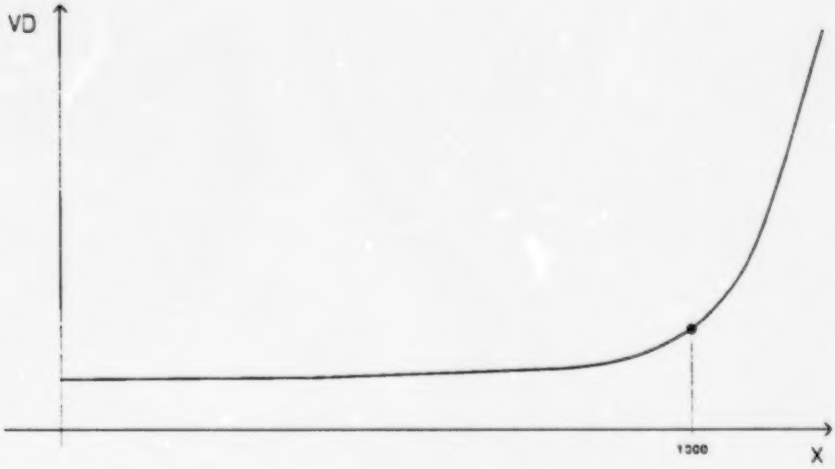


Figure 5 Volume-Delay Test Function

Tables 1-6 give the results for the derivative at $x = 1$ (the second derivative is given as well) for different choices of the step h (refered to as 'delta x' in the Tables). In each table, the estimated values of the function and its first and second derivative are given, as well as the % error relative to the exact values. The values tried for N were 3,5,7 and 11. For each of these values, the degree of the polynomial was varied within the constraints $1 \leq m \leq \min{(S, n-1)}$ as the algorithm of Casaletto and Rice does not permit a polynomial of degree higher than S. In these tables the computational time required (in CPU seconds on a 68020 supermicrocomputer) to execute one thousand estimations is given as well. Tables 7 to 11 give the results for the derivative at x = 2, while Tables 12 to 17 give the results for the derivative at x = 1.5 and Tables 18 to 21 give the results for the derivative at x = 1000.

Interpolation points:

j	x_j	y_j
-5	-4.00000	0.01832
-4	-3.00000	0.04979
-3	-2.00000	0.13534
-2	-1.00000	0.36788
-1	0.00000	1.00000
0	1.00000	2.71828
1	2.00000	7.38906
2	3.00000	20.08554
3	4.00000	54.59815
4	5.00000	148.41316
5	6.00000	403.42880

Function value : 2.7182817

First derivative : 2.7182817

Second derivative : 2.7182817

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	3.7024460	36.2	0.4	3.1945279	17.5	0.6	0.0000000	100.0	0.1
3	2	2.7182817	0.0	0.7	3.1945279	17.5	0.6	2.9524927	8.6	0.9
5	1	6.3121510	132.2	0.5	4.5824370	68.6	0.8	0.0000000	100.0	0.1
5	2	2.4434059	10.1	1.2	4.5824370	68.6	0.8	3.8687446	42.3	1.2
5	3	2.4434059	10.1	1.1	2.6162326	3.8	1.2	3.8687446	42.3	1.2
5	4	2.7182820	0.0	1.2	2.6162326	3.8	1.3	2.6852524	1.2	1.6
7	1	12.3277493	353.5	0.6	7.4718862	174.9	0.9	0.0000000	100.0	0.1
7	2	1.0121710	62.8	1.4	7.4718862	174.9	0.9	5.6577883	108.1	1.4
7	3	1.0121710	62.8	1.4	1.9582002	28.0	1.7	5.6577883	108.1	1.4
7	4	2.7936754	2.8	1.9	1.9582002	28.0	1.7	2.3422105	13.8	2.3
7	5	2.7936754	2.8	2.0	2.7418573	0.9	3.1	2.3422105	13.8	2.3
11	1	58.0185700	*****	0.7	25.6338062	843.0	1.3	0.0000000	100.0	0.1
11	2	-19.6311493	822.2	1.8	25.6338062	843.0	1.3	15.5299454	471.3	1.9
11	3	-19.6311493	822.2	1.9	-7.1136270	361.7	2.2	15.5299454	471.3	1.9
11	4	6.1922770	127.8	2.6	-7.1136270	361.7	2.2	-2.4029891	188.4	2.9
11	5	6.1922770	127.8	2.6	4.1926122	54.2	4.2	-2.4029891	188.4	2.9

Table 1 f(x)=e^x, step h=1

Interpolation points:

j	x_j	y_j
-5	0.50000	1.64872
-4	0.60000	1.82212
-3	0.70000	2.01375
-2	0.80000	2.22554
-1	0.90000	2.45960
0	1.00000	2.71828
1	1.10000	3.00417
2	1.20000	3.32012
3	1.30000	3.66930
4	1.40000	4.05520
5	1.50000	4.48169

Function value : 2.7182817
First derivative : 2.7182817
Second derivative : 2.7182817

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	2.7273502	0.3	0.5	2.7228153	0.2	0.6	0.0000000	100.0	0.1
3	2	2.7182817	0.0	0.7	2.7228153	0.2	0.6	2.7205403	0.1	0.8
5	1	2.7455418	1.0	0.5	2.7337153	0.6	0.8	0.0000000	100.0	0.1
5	2	2.7182586	0.0	1.2	2.7337153	0.6	0.8	2.7283392	0.4	1.2
5	3	2.7182586	0.0	1.1	2.7182734	0.0	1.3	2.7283392	0.4	1.2
5	4	2.7182820	0.0	1.2	2.7182734	0.0	1.3	2.7182670	0.0	1.6
7	1	2.7729652	2.0	0.6	2.7501235	1.2	1.0	0.0000000	100.0	0.1
7	2	2.7181649	0.0	1.4	2.7501235	1.2	1.0	2.7400265	0.8	1.4
7	3	2.7181649	0.0	1.4	2.7182260	0.0	1.7	2.7400265	0.8	1.4
7	4	2.7182817	0.0	2.0	2.7182260	0.0	1.7	2.7182841	0.0	2.3
7	5	2.7182817	0.0	1.9	2.7182822	0.0	3.1	2.7182841	0.0	2.3
11	1	2.8562262	5.1	0.7	2.7997737	3.0	1.3	0.0000000	100.0	0.1
11	2	2.7174575	0.0	1.8	2.7997737	3.0	1.3	2.7753673	2.1	1.9
11	3	2.7174575	0.0	1.8	2.7178466	0.0	2.2	2.7753673	2.1	1.9
11	4	2.7182837	0.0	2.5	2.7178466	0.0	2.2	2.7180262	0.0	2.9
11	5	2.7182837	0.0	2.5	2.7182996	0.0	4.2	2.7180262	0.0	2.9

Table 2 f(x)=e^x, step h=0.1

Interpolation points:

j	x_j	y_j
-5	0.95000	2.58571
-4	0.96000	2.61170
-3	0.97000	2.63794
-2	0.98000	2.66446
-1	0.99000	2.69123
0	1.00000	2.71828
1	1.01000	2.74560
2	1.02000	2.77319
3	1.03000	2.80107
4	1.04000	2.82922
5	1.05000	2.85765

Function value : 2.7182817
First derivative : 2.7182817
Second derivative : 2.7182817

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	2.7183723	0.0	0.5	2.7183175	0.0	0.6	0.0000000	100.0	0.1
3	2	2.7182817	0.0	0.7	2.7183175	0.0	0.6	2.7160647	0.1	0.9
5	1	2.7185538	0.0	0.5	2.7184322	0.0	0.8	0.0000000	100.0	0.1
5	2	2.7182817	0.0	1.2	2.7184322	0.0	0.8	2.7194555	0.0	1.2
5	3	2.7182817	0.0	1.1	2.7182698	0.0	1.3	2.7194555	0.0	1.2
5	4	2.7182820	0.0	1.2	2.7182698	0.0	1.3	2.7191327	0.0	1.6
7	1	2.7188258	0.0	0.6	2.7185960	0.0	1.0	0.0000000	100.0	0.1
7	2	2.7182820	0.0	1.4	2.7185960	0.0	0.9	2.7184865	0.0	1.4
7	3	2.7182820	0.0	1.4	2.7182760	0.0	1.7	2.7184865	0.0	1.4
7	4	2.7182817	0.0	2.0	2.7182760	0.0	1.7	2.7206590	0.1	2.3
7	5	2.7182817	0.0	2.0	2.7182686	0.0	3.1	2.7206590	0.1	2.3
11	1	2.7196412	0.1	0.7	2.7190874	0.0	1.3	0.0000000	100.0	0.1
11	2	2.7182817	0.0	1.9	2.7190874	0.0	1.3	2.7188704	0.0	1.9
11	3	2.7182817	0.0	1.8	2.7182779	0.0	2.3	2.7188704	0.0	1.9
11	4	2.7182820	0.0	2.5	2.7182779	0.0	2.3	2.7184455	0.0	2.9
11	5	2.7182820	0.0	2.5	2.7182920	0.0	4.2	2.7184455	0.0	2.9

Table 3 f(x)=e^x, step h=0.01

Interpolation points:

j	x_j	y_j
-5	0.99500	2.70472
-4	0.99600	2.70743
-3	0.99700	2.71014
-2	0.99800	2.71285
-1	0.99900	2.71556
0	1.00000	2.71828
1	1.00100	2.72100
2	1.00200	2.72372
3	1.00300	2.72645
4	1.00400	2.72918
5	1.00500	2.73191

Function value : 2.7182817
First derivative : 2.7182817
Second derivative : 2.7182817

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	2.7182827	0.0	0.5	2.7183294	0.0	0.6	0.0000000	100.0	0.1
3	2	2.7182817	0.0	0.7	2.7183294	0.0	0.6	2.5431314	6.4	0.9
5	1	2.7182844	0.0	0.5	2.7182341	0.0	0.8	0.0000000	100.0	0.1
5	2	2.7182817	0.0	1.1	2.7182341	0.0	0.8	2.7126732	0.2	1.2
5	3	2.7182817	0.0	1.1	2.7183692	0.0	1.3	2.7126732	0.2	1.2
5	4	2.7182820	0.0	1.2	2.7183692	0.0	1.3	2.7126732	0.2	1.6
7	1	2.7182870	0.0	0.6	2.7182951	0.0	1.0	0.0000000	100.0	0.1
7	2	2.7182820	0.0	1.4	2.7182951	0.0	1.0	2.7772608	2.2	1.4
7	3	2.7182820	0.0	1.4	2.7182026	0.0	1.7	2.7772608	2.2	1.4
7	4	2.7182820	0.0	2.0	2.7182026	0.0	1.7	2.5052104	7.8	2.3
7	5	2.7182820	0.0	1.9	2.7184007	0.0	3.1	2.5052104	7.8	2.3
11	1	2.7182953	0.0	0.7	2.7182925	0.0	1.3	0.0000000	100.0	0.1
11	2	2.7182817	0.0	1.8	2.7182925	0.0	1.3	2.7189965	0.0	1.9
11	3	2.7182817	0.0	1.8	2.7182693	0.0	2.3	2.7189965	0.0	1.9
11	4	2.7182815	0.0	2.5	2.7182693	0.0	2.3	2.7661929	1.8	2.9
11	5	2.7182815	0.0	2.5	2.7182932	0.0	4.3	2.7661929	1.8	2.9

Table 4 f(x)=e^x, step h=0.001

Interpolation points:

j	x_j	y_j
-5	0.99950	2.71692
-4	0.99960	2.71719
-3	0.99970	2.71747
-2	0.99980	2.71774
-1	0.99990	2.71801
0	1.00000	2.71828
1	1.00010	2.71855
2	1.00020	2.71883
3	1.00030	2.71910
4	1.00040	2.71937
5	1.00050	2.71964

Function value : 2.7182817
First derivative : 2.7182817
Second derivative : 2.7182817

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	2.7182817	0.0	0.5	2.7191641	0.0	0.7	0.0000000	100.0	0.1
3	2	2.7182817	0.0	0.8	2.7191641	0.0	0.6	0.0000000	100.0	0.9
5	1	2.7182817	0.0	0.5	2.7182102	0.0	0.8	0.0000000	100.0	0.1
5	2	2.7182815	0.0	1.2	2.7182102	0.0	0.8	19.3762417	612.8	1.3
5	3	2.7182815	0.0	1.2	2.7195611	0.1	1.3	19.3762417	612.8	1.3
5	4	2.7182820	0.0	1.2	2.7195611	0.1	1.3	17.2233257	533.6	1.7
7	1	2.7182820	0.0	0.6	2.7183125	0.0	1.0	0.0000000	100.0	0.1
7	2	2.7182820	0.0	1.4	2.7183125	0.0	1.0	12.9174948	375.2	1.5
7	3	2.7182820	0.0	1.4	2.7183125	0.0	1.8	12.9174948	375.2	1.5
7	4	2.7182815	0.0	2.0	2.7183125	0.0	1.8	5.0104222	84.3	2.4
7	5	2.7182815	0.0	2.0	2.7197995	0.1	3.2	5.0104222	84.3	2.4
11	1	2.7182820	0.0	0.7	2.7181668	0.0	1.3	0.0000000	100.0	0.1
11	2	2.7182817	0.0	1.9	2.7181668	0.0	1.3	2.5292995	7.0	2.0
11	3	2.7182817	0.0	1.9	2.7182167	0.0	2.3	2.5292995	7.0	2.0
11	4	2.7182817	0.0	2.6	2.7182167	0.0	2.3	12.3333473	353.7	3.0
11	5	2.7182817	0.0	2.6	2.7186694	0.0	5.0	12.3333473	353.7	3.0

Table 5 f(x)=e^x, step h=0.0001

Interpolation points:

j	x_j	y_j
-5	0.99995	2.71815
-4	0.99996	2.71817
-3	0.99997	2.71820
-2	0.99998	2.71823
-1	0.99999	2.71825
0	1.00000	2.71828
1	1.00001	2.71831
2	1.00002	2.71834
3	1.00003	2.71836
4	1.00004	2.71839
5	1.00005	2.71842

Function value : 2.7182817
First derivative : 2.7182817
Second derivative : 2.7182817

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	2.7182817	0.0	0.5	2.7179718	0.0	0.6	0.0000000	100.0	0.2
3	2	2.7182817	0.0	0.8	2.7179718	0.0	0.6	0.0000000	100.0	0.9
5	1	2.7182817	0.0	0.5	2.7227404	0.2	0.8	0.0000000	100.0	0.1
5	2	2.7182815	0.0	1.2	2.7227404	0.2	0.8	0.0000000	100.0	1.3
5	3	2.7182815	0.0	1.2	2.7159851	0.1	1.3	0.0000000	100.0	1.2
5	4	2.7182820	0.0	1.4	2.7159851	0.1	1.6	1722.3324000	*****	1.8
7	1	2.7182820	0.0	0.6	2.7196748	0.1	1.0	0.0000000	100.0	0.1
7	2	2.7182820	0.0	1.4	2.7196748	0.1	1.0	645.8746950	*****	1.4
7	3	2.7182820	0.0	1.4	2.7243109	0.2	1.8	645.8746950	*****	1.4
7	4	2.7182815	0.0	2.0	2.7243109	0.2	1.8	-1002.0844100	*****	2.3
7	5	2.7182815	0.0	2.0	2.7143958	0.1	3.2	-1002.0844100	*****	2.3
11	1	2.7182820	0.0	0.7	2.7192724	0.0	1.3	0.0000000	100.0	0.1
11	2	2.7182817	0.0	1.9	2.7192724	0.0	1.4	-126.4649810	*****	1.9
11	3	2.7182817	0.0	1.9	2.7216632	0.1	2.4	-126.4649810	*****	2.0
11	4	2.7182815	0.0	2.6	2.7216632	0.1	2.3	704.7626950	*****	3.0
11	5	2.7182815	0.0	2.8	2.7202220	0.1	4.7	704.7626950	*****	3.1

Table 6 $f(x)=e^x$, step h=0.00001

Interpolation points:

j	x_j	y_j
-5	-3.00000	-0.14112
-4	-2.00000	-0.90930
-3	-1.00000	-0.84147
-2	0.00000	0.00000
-1	1.00000	0.84147
0	2.00000	0.90930
1	3.00000	0.14112
2	4.00000	-0.75680
3	5.00000	-0.95892
4	6.00000	-0.27942
5	7.00000	0.65699

Function value	:	0.9092974
First derivative	:	-0.4161468
Second derivative	:	-0.9092974

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	0.6306295	30.6	0.5	-0.3501755	15.9	0.7	0.0000000	100.0	0.1
3	2	0.9092974	0.0	0.7	-0.3501755	15.9	0.6	-0.8360038	8.1	0.9
5	1	0.2270172	75.0	0.5	-0.2213956	46.8	0.8	0.0000000	100.0	0.1
5	2	0.8434159	7.2	1.1	-0.2213956	46.8	0.8	-0.6163987	32.2	1.2
5	3	0.8434159	7.2	1.1	-0.4038338	3.0	1.3	-0.6163987	32.2	1.2
5	4	0.9092974	0.0	1.1	-0.4038338	3.0	1.3	-0.9000553	1.0	1.6
7	1	-0.0950442	110.5	0.6	-0.0916541	78.0	1.0	0.0000000	100.0	0.1
7	2	0.6471910	28.8	1.4	-0.0916541	78.0	1.0	-0.3711176	59.2	1.4
7	3	0.6471910	28.8	1.4	-0.3521514	15.4	1.7	-0.3711176	59.2	1.4
7	4	0.8940017	1.7	2.0	-0.3521514	15.4	1.8	-0.8304597	8.7	2.4
7	5	0.8940017	1.7	2.0	-0.4137004	0.6	3.2	-0.8304597	8.7	2.3
11	1	-0.1216505	113.4	0.7	0.0358522	108.6	1.2	0.0000000	100.0	0.1
11	2	0.0064306	99.3	1.9	0.0358522	108.6	1.3	-0.0256162	97.2	1.9
11	3	0.0064306	99.3	1.9	-0.1368895	67.1	2.2	-0.0256162	97.2	1.9
11	4	0.6604054	27.4	2.6	-0.1368895	67.1	2.3	-0.4797654	47.2	2.9
11	5	0.6604054	27.4	2.6	-0.3537153	15.0	4.3	-0.4797654	47.2	2.9

Table 7 f(x)=sin x, step h=1

Interpolation points:

j	x_j	y_j
-5	1.50000	0.99749
-4	1.60000	0.99957
-3	1.70000	0.99166
-2	1.80000	0.97385
-1	1.90000	0.94630
0	2.00000	0.90930
1	2.10000	0.86321
2	2.20000	0.80850
3	2.30000	0.74571
4	2.40000	0.67546
5	2.50000	0.59847

Function value : 0.9092974
First derivative : -0.4161468
Second derivative : -0.9092974

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	0.9062690	0.3	0.5	-0.4154533	0.2	0.6	0.0000000	100.0	0.1
3	2	0.9092974	0.0	0.8	-0.4154533	0.2	0.6	-0.9085337	0.1	0.9
5	1	0.9002302	1.0	0.5	-0.4137931	0.6	0.9	0.0000000	100.0	0.1
5	2	0.9092896	0.0	1.2	-0.4137931	0.6	0.9	-0.9059458	0.4	1.3
5	3	0.9092896	0.0	1.2	-0.4161451	0.0	1.3	-0.9059458	0.4	1.3
5	4	0.9092974	0.0	1.2	-0.4161451	0.0	1.3	-0.9092947	0.0	1.7
7	1	0.8912173	2.0	0.6	-0.4113114	1.2	0.9	0.0000000	100.0	0.1
7	2	0.9092587	0.0	1.4	-0.4113114	1.2	1.0	-0.9020674	0.8	1.5
7	3	0.9092587	0.0	1.6	-0.4161380	0.0	1.9	-0.9020674	0.8	1.7
7	4	0.9092974	0.0	2.3	-0.4161380	0.0	1.8	-0.9092888	0.0	2.5
7	5	0.9092974	0.0	2.2	-0.4161464	0.0	3.3	-0.9092888	0.0	2.4
11	1	0.8645022	4.9	0.7	-0.4039299	2.9	1.3	0.0000000	100.0	0.1
11	2	0.9090276	0.0	1.9	-0.4039299	2.9	1.3	-0.8905056	2.1	1.9
11	3	0.9090276	0.0	1.9	-0.4160813	0.0	2.3	-0.8905056	2.1	1.9
11	4	0.9092969	0.0	2.6	-0.4160813	0.0	2.3	-0.9092125	0.0	3.0
11	5	0.9092969	0.0	2.6	-0.4161489	0.0	4.4	-0.9092125	0.0	3.0

Table 8 f(x)=sin x, step h=0.1

Interpolation points:

j	x_j	y_j
-5	1.95000	0.92896
-4	1.96000	0.92521
-3	1.97000	0.92137
-2	1.98000	0.91744
-1	1.99000	0.91341
0	2.00000	0.90930
1	2.01000	0.90509
2	2.02000	0.90079
3	2.03000	0.89641
4	2.04000	0.89193
5	2.05000	0.88736

Function value	:	0.9092974
First derivative	:	-0.4161468
Second derivative	:	-0.9092974

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	0.9092671	0.0	0.5	-0.4161358	0.0	0.6	0.0000000	100.0	0.1
3	2	0.9092974	0.0	0.8	-0.4161358	0.0	0.6	-0.9104412	0.1	1.0
5	1	0.9092065	0.0	0.7	-0.4161215	0.0	1.0	0.0000000	100.0	0.2
5	2	0.9092974	0.0	1.3	-0.4161215	0.0	0.9	-0.9092302	0.0	1.4
5	3	0.9092974	0.0	1.2	-0.4161418	0.0	1.3	-0.9092302	0.0	1.4
5	4	0.9092974	0.0	1.2	-0.4161418	0.0	1.3	-0.9110064	0.2	1.7
7	1	0.9091156	0.0	0.6	-0.4160968	0.0	1.0	0.0000000	100.0	0.1
7	2	0.9092974	0.0	1.5	-0.4160968	0.0	1.0	-0.9090686	0.0	1.5
7	3	0.9092974	0.0	1.4	-0.4161443	0.0	1.8	-0.9090686	0.0	1.5
7	4	0.9092974	0.0	2.0	-0.4161443	0.0	1.8	-0.9097673	0.1	2.5
7	5	0.9092974	0.0	2.0	-0.4161412	0.0	3.3	-0.9097673	0.1	2.4
11	1	0.9088429	0.1	0.7	-0.4160227	0.0	1.3	0.0000000	100.0	0.1
11	2	0.9092973	0.0	1.9	-0.4160227	0.0	1.3	-0.9090934	0.0	1.9
11	3	0.9092973	0.0	1.9	-0.4161451	0.0	2.4	-0.9090934	0.0	1.9
11	4	0.9092973	0.0	2.6	-0.4161451	0.0	2.3	-0.9091438	0.0	3.0
11	5	0.9092973	0.0	2.7	-0.4161468	0.0	4.4	-0.9091438	0.0	3.0

Table 9 f(x)=sin x, step h=0.01

Interpolation points:

j	x_j	y_j
-5	1.99500	0.91137
-4	1.99600	0.91095
-3	1.99700	0.91054
-2	1.99800	0.91013
-1	1.99900	0.90971
0	2.00000	0.90930
1	2.00100	0.90888
2	2.00200	0.90846
3	2.00300	0.90804
4	2.00400	0.90763
5	2.00500	0.90721

Function value : 0.9092974
First derivative : -0.4161468
Second derivative : -0.9092974

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	0.9092970	0.0	0.5	-0.4161596	0.0	0.7	0.0000000	100.0	0.2
3	2	0.9092974	0.0	0.8	-0.4161596	0.0	0.6	-1.0172526	11.9	0.9
5	1	0.9092964	0.0	0.5	-0.4161477	0.0	0.8	0.0000000	100.0	0.1
5	2	0.9092974	0.0	1.2	-0.4161477	0.0	0.8	-0.9203713	1.2	1.3
5	3	0.9092974	0.0	1.1	-0.4161646	0.0	1.3	-0.9203713	1.2	1.3
5	4	0.9092974	0.0	1.2	-0.4161646	0.0	1.3	-1.0549284	16.0	1.8
7	1	0.9092955	0.0	0.8	-0.4161490	0.0	1.2	0.0000000	100.0	0.2
7	2	0.9092974	0.0	1.8	-0.4161490	0.0	1.1	-0.9203713	1.2	1.6
7	3	0.9092974	0.0	1.5	-0.4161490	0.0	1.9	-0.9203713	1.2	1.5
7	4	0.9092974	0.0	2.1	-0.4161490	0.0	1.9	-0.8893498	2.2	2.4
7	5	0.9092974	0.0	2.0	-0.4161676	0.0	3.2	-0.8893498	2.2	2.5
11	1	0.9092928	0.0	0.8	-0.4161477	0.0	1.5	0.0000000	100.0	0.1
11	2	0.9092974	0.0	2.0	-0.4161477	0.0	1.4	-0.9042245	0.6	2.1
11	3	0.9092974	0.0	1.9	-0.4161409	0.0	2.3	-0.9042245	0.6	2.1
11	4	0.9092975	0.0	2.6	-0.4161409	0.0	2.5	-0.8897627	2.1	3.1
11	5	0.9092975	0.0	2.7	-0.4161652	0.0	4.7	-0.8897627	2.1	3.0

Table 10 f(x)=sin x, step h=0.001

Interpolation points:

j	x_j	y_j
-5	1.99950	0.90951
-4	1.99960	0.90946
-3	1.99970	0.90942
-2	1.99980	0.90938
-1	1.99990	0.90934
0	2.00000	0.90930
1	2.00010	0.90926
2	2.00020	0.90921
3	2.00030	0.90917
4	2.00040	0.90913
5	2.00050	0.90909

Function value : 0.9092974
First derivative : -0.4161468
Second derivative : -0.9092974

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	0.9092975	0.0	0.5	-0.4157424	0.1	0.6	0.0000000	100.0	0.1
3	2	0.9092974	0.0	0.7	-0.4157424	0.1	0.6	0.0000000	100.0	0.9
5	1	0.9092973	0.0	0.5	-0.4163384	0.1	0.9	0.0000000	100.0	0.1
5	2	0.9092975	0.0	1.2	-0.4163384	0.1	0.8	-4.8440604	432.7	1.2
5	3	0.9092975	0.0	1.2	-0.4154940	0.2	1.3	-4.8440604	432.7	1.3
5	4	0.9092974	0.0	1.2	-0.4154940	0.2	1.4	19.3762398	*****	1.7
7	1	0.9092974	0.0	0.6	-0.4162107	0.0	1.0	0.0000000	100.0	0.1
7	2	0.9092975	0.0	1.4	-0.4162107	0.0	1.0	1.6146868	277.6	1.4
7	3	0.9092975	0.0	1.5	-0.4163266	0.0	1.9	1.6146868	277.6	1.4
7	4	0.9092975	0.0	2.2	-0.4163266	0.0	1.8	-11.2734499	*****	2.3
7	5	0.9092975	0.0	2.1	-0.4153351	0.2	3.3	-11.2734499	*****	2.4
11	1	0.9092974	0.0	0.7	-0.4161542	0.0	1.3	0.0000000	100.0	0.1
11	2	0.9092975	0.0	1.9	-0.4161542	0.0	1.4	-0.6323249	30.5	1.9
11	3	0.9092975	0.0	1.9	-0.4162758	0.0	2.4	-0.6323249	30.5	1.9
11	4	0.9092973	0.0	2.6	-0.4162758	0.0	2.4	-1.7619067	93.8	2.9
11	5	0.9092973	0.0	2.6	-0.4162418	0.0	4.4	-1.7619067	93.8	3.0

Table 11 f(x)=sin x, step h=0.0001

Interpolation points:

j	x_j	y_j
-5	-3.50000	-0.37459
-4	-2.50000	0.74702
-3	-1.50000	-14.10142
-2	-0.50000	-0.54630
-1	0.50000	0.54630
0	1.50000	14.10142
1	2.50000	-0.74702
2	3.50000	0.37459
3	4.50000	4.63733
4	5.50000	-0.99558
5	6.50000	0.22028

Function value	:	14.1014204
First derivative	:	199.8500210
Second derivative	:	5636.3378900

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	4.6335669	67.1	0.5	-0.6466624	100.3	0.6	0.0000000	100.0	0.1
3	2	14.1014204	0.0	0.7	-0.6466624	100.3	0.6	-28.4035606	100.5	0.9
5	1	2.7457967	80.5	0.5	0.0548451	100.0	0.8	0.0000000	100.0	0.1
5	2	6.7951617	51.8	1.2	0.0548451	100.0	0.8	-4.0493650	100.1	1.3
5	3	6.7951617	51.8	1.2	-0.9389571	100.5	1.3	-4.0493650	100.1	1.2
5	4	14.1014195	0.0	1.2	-0.9389571	100.5	1.3	-35.5068665	100.6	1.7
7	1	0.6092705	95.7	0.6	2.0273108	99.0	1.0	0.0000000	100.0	0.1
7	2	5.5199356	60.9	1.4	2.0273108	99.0	1.0	-2.4553323	100.0	1.4
7	3	5.5199356	60.9	1.4	-1.6887535	100.8	1.7	-2.4553323	100.0	1.4
7	4	7.7491913	45.0	2.0	-1.6887535	100.8	1.7	-6.6042256	100.1	2.3
7	5	7.7491913	45.0	2.0	-0.7958142	100.4	3.2	-6.6042256	100.1	2.3
11	1	0.3510930	97.5	0.7	0.4797145	99.8	1.3	0.0000000	100.0	0.1
11	2	1.8956076	86.6	1.9	0.4797145	99.8	1.3	-0.3089029	100.0	1.9
11	3	1.8956076	86.6	1.9	1.8179126	99.1	2.3	-0.3089029	100.0	1.9
11	4	4.8605185	65.5	2.6	1.8179126	99.1	2.3	-2.3678684	100.0	2.9
11	5	4.8605185	65.5	2.7	1.5738267	99.2	4.3	-2.3678684	100.0	3.0

Table 12 f(x)=tg x, step h=1

Interpolation points:

j	x_j	y_j
-5	1.00000	1.55741
-4	1.10000	1.96476
-3	1.20000	2.57215
-2	1.30000	3.60210
-1	1.40000	5.79788
0	1.50000	14.10142
1	1.60000	-34.23251
2	1.70000	-7.69660
3	1.80000	-4.28626
4	1.90000	-2.92710
5	2.00000	-2.18504

Function value	:	14.1014204
First derivative	:	199.8500210
Second derivative	:	5636.3378900

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	-4.7777343	133.9	0.5	-200.1519320	200.2	0.7	0.0000000	100.0	0.2
3	2	14.1014204	0.0	0.7	-200.1519320	200.2	0.6	-5663.7461000	200.5	0.9
5	1	-3.6855397	126.1	0.5	-62.6277886	131.3	0.8	0.0000000	100.0	0.1
5	2	-2.5487962	118.1	1.2	-62.6277886	131.3	0.9	-113.6744310	102.0	1.4
5	3	-2.5487962	118.1	1.2	-257.4536740	228.8	1.3	-113.6744310	102.0	1.3
5	4	14.1014195	0.0	1.3	-257.4536740	228.8	1.3	-7282.5170900	229.2	1.7
7	1	-2.8774014	120.4	0.6	-29.7153664	114.9	1.0	0.0000000	100.0	0.1
7	2	-3.8453846	127.3	1.4	-29.7153664	114.9	1.2	48.3991432	99.1	1.8
7	3	-3.8453846	127.3	1.5	-116.1861110	158.1	1.8	48.3991432	99.1	1.5
7	4	-0.7404599	105.3	2.1	-116.1861110	158.1	2.1	-529.4617310	109.4	2.5
7	5	-0.7404599	105.3	2.2	-284.4229130	242.3	3.2	-529.4617310	109.4	3.1
11	1	-1.9756163	114.0	0.9	-11.0438814	105.5	1.7	0.0000000	100.0	0.2
11	2	-3.4440312	124.4	2.4	-11.0438814	105.5	1.5	29.3682938	99.5	2.1
11	3	-3.4440312	124.4	2.5	-41.7567864	120.9	2.5	29.3682938	99.5	2.1
11	4	-3.7113581	126.3	3.0	-41.7567864	120.9	2.5	47.9326668	99.1	3.7
11	5	-3.7113581	126.3	3.4	-96.8588943	148.5	4.9	47.9326668	99.1	3.0

Table 13 f(x)=tg x, step h=0.1

Interpolation points:

j	x_j	y_j
-5	1.45000	8.23810
-4	1.46000	8.98861
-3	1.47000	9.88738
-2	1.48000	10.98338
-1	1.49000	12.34986
0	1.50000	14.10142
1	1.51000	16.42809
2	1.52000	19.66952
3	1.53000	24.49839
4	1.54000	32.46110
5	1.55000	48.07837

Function value : 14.1014204

First derivative : 199.8500210

Second derivative : 5636.3378900

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	14.2931223	1.4	0.5	203.9115450	2.0	0.6	0.0000000	100.0	0.1
3	2	14.1014204	0.0	0.8	203.9115450	2.0	0.6	5751.0581100	2.0	0.9
5	1	14.7064543	4.3	0.6	214.5050960	7.3	0.8	0.0000000	100.0	0.1
5	2	14.0885944	0.1	1.2	214.5050960	7.3	0.8	6178.6108400	9.6	1.2
5	3	14.0885944	0.1	1.2	199.4975590	0.2	1.3	6178.6108400	9.6	1.2
5	4	14.1014195	0.0	1.2	199.4975590	0.2	1.3	5626.3564500	0.2	1.7
7	1	15.4168625	9.3	0.6	233.1555630	16.7	1.0	0.0000000	100.0	0.1
7	2	14.0268946	0.5	1.4	233.1555630	16.7	1.0	6949.8388700	23.3	1.4
7	3	14.0268946	0.5	1.4	197.2485500	1.3	1.8	6949.8388700	23.3	1.5
7	4	14.1037817	0.0	2.0	197.2485500	1.3	2.1	5518.8754900	2.1	2.6
7	5	14.1037817	0.0	2.1	199.9269260	0.0	3.2	5518.8754900	2.1	2.3
11	1	18.6985645	32.6	0.7	325.7953490	63.0	1.3	0.0000000	100.0	0.1
11	2	13.1609640	6.7	1.9	325.7953490	63.0	1.3	11075.2070000	96.5	2.0
11	3	13.1609640	6.7	1.9	161.1921390	19.3	2.3	11075.2070000	96.5	1.9
11	4	14.2508917	1.1	2.6	161.1921390	19.3	2.3	3506.2419400	37.8	3.0
11	5	14.2508917	1.1	2.6	207.4321900	3.8	4.3	3506.2419400	37.8	3.0

Table 14 f(x)=tg x, step h=0.01

Interpolation points:

j	x_j	y_j
-5	1.49500	13.16798
-4	1.49600	13.34471
-3	1.49700	13.52620
-2	1.49800	13.71269
-1	1.49900	13.90434
0	1.50000	14.10142
1	1.50100	14.30414
2	1.50200	14.51272
3	1.50300	14.72746
4	1.50400	14.94860
5	1.50500	15.17648

Function value	:	14.1014204
First derivative	:	199.8500210
Second derivative	:	5636.3378900

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	14.1032991	0.0	0.4	199.8992000	0.0	0.7	0.0000000	100.0	0.2
3	2	14.1014204	0.0	0.9	199.8992000	0.0	0.8	5637.6132800	0.0	1.4
5	1	14.1070604	0.0	0.9	199.9853970	0.1	1.1	0.0000000	100.0	0.2
5	2	14.1014185	0.0	1.2	199.9853970	0.1	1.0	5640.8100600	0.1	1.4
5	3	14.1014185	0.0	1.3	199.8632810	0.0	1.5	5640.8100600	0.1	1.8
5	4	14.1014195	0.0	1.4	199.8632810	0.0	1.7	5634.9541000	0.0	2.3
7	1	14.1127090	0.1	0.7	200.1301270	0.1	1.3	0.0000000	100.0	0.1
7	2	14.1014137	0.0	1.7	200.1301270	0.1	1.1	5647.2690400	0.2	1.7
7	3	14.1014137	0.0	1.5	199.8506780	0.0	1.9	5647.2690400	0.2	1.6
7	4	14.1014204	0.0	2.8	199.8506780	0.0	2.3	5635.7216800	0.0	2.8
7	5	14.1014204	0.0	2.3	199.8656620	0.0	3.5	5635.7216800	0.0	2.6
11	1	14.1297035	0.2	0.8	200.5609590	0.4	1.5	0.0000000	100.0	0.1
11	2	14.1013784	0.0	2.1	200.5609590	0.4	1.5	5664.6181700	0.5	2.1
11	3	14.1013784	0.0	2.1	199.8485110	0.0	2.6	5664.6181700	0.5	2.1
11	4	14.1014204	0.0	3.0	199.8485110	0.0	2.6	5635.9858400	0.0	3.3
11	5	14.1014204	0.0	2.8	199.8558200	0.0	4.9	5635.9858400	0.0	3.0

Table 15 f(x)=tg x, step h=0.001

Interpolation points:

j	x_j	y_j
-5	1.49950	14.00220
-4	1.49960	14.02194
-3	1.49970	14.04171
-2	1.49980	14.06155
-1	1.49990	14.08146
0	1.50000	14.10142
1	1.50010	14.12144
2	1.50020	14.14151
3	1.50030	14.16164
4	1.50040	14.18180
5	1.50050	14.20205

Function value : 14.1014204
First derivative : 199.8500210
Second derivative : 5636.3378900

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	14.1014395	0.0	0.5	199.8806150	0.0	0.7	0.0000000	100.0	0.1
3	2	14.1014204	0.0	0.8	199.8806150	0.0	0.7	5696.6147500	1.1	0.9
5	1	14.1014767	0.0	0.6	199.8844150	0.0	0.9	0.0000000	100.0	0.2
5	2	14.1014204	0.0	1.2	199.8844150	0.0	0.9	5580.3574200	1.0	1.4
5	3	14.1014204	0.0	1.2	199.8790130	0.0	1.3	5580.3574200	1.0	1.3
5	4	14.1014195	0.0	1.4	199.8790130	0.0	1.6	5339.2304700	5.3	1.8
7	1	14.1015329	0.0	0.6	199.8860630	0.0	1.0	0.0000000	100.0	0.2
7	2	14.1014195	0.0	1.5	199.8860630	0.0	1.0	5632.0278300	0.1	1.6
7	3	14.1014195	0.0	1.4	199.8823400	0.0	1.9	5632.0278300	0.1	2.0
7	4	14.1014204	0.0	2.2	199.8823400	0.0	1.9	5531.5063500	1.9	2.4
7	5	14.1014204	0.0	2.0	199.8783720	0.0	3.6	5531.5063500	1.9	2.5
11	1	14.1017008	0.0	0.8	199.8513790	0.0	1.3	0.0000000	100.0	0.2
11	2	14.1014195	0.0	1.9	199.8513790	0.0	1.4	5635.2793000	0.0	2.0
11	3	14.1014196	0.0	2.1	199.8840940	0.0	2.9	5635.2793000	0.0	2.0
11	4	14.1014214	0.0	2.7	199.8840940	0.0	2.4	5624.0063500	0.2	3.1
11	5	14.1014214	0.0	2.7	199.9275670	0.0	5.2	5624.0063500	0.2	3.7

Table 16 f(x)=tg x, step h=0.0001

Interpolation points:

j	x_j	y_j
-5	1.49995	14.09144
-4	1.49996	14.09342
-3	1.49997	14.09542
-2	1.49998	14.09742
-1	1.49999	14.09942
0	1.50000	14.10142
1	1.50001	14.10342
2	1.50002	14.10542
3	1.50003	14.10743
4	1.50004	14.10943
5	1.50005	14.11141

Function value : 14.1014204
First derivative : 199.8500210
Second derivative : 5636.3378900

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	14.1014204	0.0	0.5	200.1285550	0.1	0.6	0.0000000	100.0	0.2
3	2	14.1014204	0.0	0.9	200.1285550	0.1	0.8	-20345.0527000	461.0	0.9
5	1	14.1014204	0.0	0.7	200.1094970	0.1	1.1	0.0000000	100.0	0.2
5	2	14.1014195	0.0	1.5	200.1094970	0.1	1.0	7750.4956100	37.5	1.6
5	3	14.1014195	0.0	1.3	200.1365200	0.1	1.3	7750.4956100	37.5	1.2
5	4	14.1014195	0.0	1.2	200.1365200	0.1	1.3	-25834.9863000	558.4	1.7
7	1	14.1014223	0.0	0.6	200.1217500	0.1	1.0	0.0000000	100.0	0.1
7	2	14.1014195	0.0	1.4	200.1217500	0.1	1.0	5166.9975600	8.3	1.5
7	3	14.1014195	0.0	1.4	200.1031950	0.1	1.8	5166.9975600	8.3	1.4
7	4	14.1014204	0.0	2.1	200.1031950	0.1	1.8	6012.5063500	6.7	2.3
7	5	14.1014204	0.0	2.0	200.1428530	0.1	3.2	6012.5063500	6.7	2.4
11	1	14.1014242	0.0	0.7	199.8979340	0.0	1.3	0.0000000	100.0	0.1
11	2	14.1014195	0.0	1.9	199.8979340	0.0	1.3	6070.3193400	7.7	1.9
11	3	14.1014195	0.0	1.9	200.4027710	0.3	2.4	6070.3193400	7.7	2.0
11	4	14.1014185	0.0	2.6	200.4027710	0.3	2.4	8457.1523400	50.0	3.0
11	5	14.1014185	0.0	2.6	199.9552150	0.1	4.6	8457.1523400	50.0	3.0

Table 17 f(x)=tg x, step h=0.00001

Interpolation points:

j	x_j	y_j
-5	500.00000	12.00000
-4	600.00000	21.99350
-3	700.00000	37.91588
-2	800.00000	61.65625
-1	900.00000	95.33305
0	1000.00000	141.28809
1	1100.00000	202.08116
2	1200.00000	280.48563
3	1300.00000	379.48410
4	1400.00000	502.26538
5	1500.00000	652.22064

Function value : 141.2880860
First derivative : 0.5292947
Second derivative : 0.0014820

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	146.2341160	3.5	0.5	0.5337405	0.8	0.6	0.0000000	100.0	0.1
3	2	141.2880860	0.0	0.7	0.5337405	0.8	0.7	0.0014838	0.1	0.9
5	1	156.1688540	10.5	0.5	0.5444068	2.9	0.8	0.0000000	100.0	0.1
5	2	141.2697910	0.0	1.2	0.5444068	2.9	1.0	0.0014899	0.5	1.3
5	3	141.2697910	0.0	1.2	0.5292963	0.0	1.3	0.0014899	0.5	1.2
5	4	141.2880860	0.0	1.2	0.5292963	0.0	1.3	0.0014820	0.0	1.7
7	1	171.1777500	21.2	0.6	0.5603970	5.9	0.9	0.0000000	100.0	0.2
7	2	141.1965480	0.1	1.4	0.5603970	5.9	1.0	0.0014991	1.1	1.4
7	3	141.1965480	0.1	1.4	0.5293038	0.0	1.8	0.0014991	1.1	1.4
7	4	141.2881010	0.0	2.0	0.5293038	0.0	1.8	0.0014820	0.0	2.3
7	5	141.2881010	0.0	2.0	0.5292948	0.0	3.2	0.0014820	0.0	2.4
11	1	216.9748990	53.6	0.7	0.6083003	14.9	1.3	0.0000000	100.0	0.1
11	2	140.6461640	0.5	1.9	0.6083003	14.9	1.3	0.0015266	3.0	1.9
11	3	140.6461640	0.5	1.9	0.5293642	0.0	2.3	0.0015266	3.0	2.0
11	4	141.2884520	0.0	2.6	0.5293642	0.0	2.3	0.0014820	0.0	2.9
11	5	141.2884520	0.0	2.6	0.5292983	0.0	4.4	0.0014820	0.0	3.0

Table 18 f(x)=vd(x), step h=100

Interpolation points:

j	x_j	y_j
-5	950.00000	116.62086
-4	960.00000	121.27368
-3	970.00000	126.06424
-2	980.00000	130.99507
-1	990.00000	136.06880
0	1000.00000	141.28809
1	1010.00000	146.65558
2	1020.00000	152.17393
3	1030.00000	157.84590
4	1040.00000	163.67416
5	1050.00000	169.66145

Function value	:	141.2880860
First derivative	:	0.5292947
Second derivative	:	0.0014820

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	141.3374790	0.0	0.5	0.5293388	0.0	0.6	0.0000000	100.0	0.1
3	2	141.2880860	0.0	0.8	0.5293388	0.0	0.7	0.0014818	0.0	0.9
5	1	141.4362950	0.1	0.5	0.5294451	0.0	0.8	0.0000000	100.0	0.1
5	2	141.2880710	0.0	1.2	0.5294451	0.0	0.8	0.0014821	0.0	1.3
5	3	141.2880710	0.0	1.2	0.5292946	0.0	1.3	0.0014821	0.0	1.3
5	4	141.2880860	0.0	1.2	0.5292946	0.0	1.3	0.0014820	0.0	1.7
7	1	141.5845030	0.2	0.6	0.5296053	0.1	1.0	0.0000000	100.0	0.1
7	2	141.2880710	0.0	1.4	0.5296053	0.1	1.0	0.0014822	0.0	1.4
7	3	141.2880710	0.0	1.4	0.5292938	0.0	1.8	0.0014822	0.0	1.4
7	4	141.2880860	0.0	2.1	0.5292938	0.0	2.0	0.0014820	0.0	2.3
7	5	141.2880860	0.0	2.0	0.5292947	0.0	3.2	0.0014820	0.0	2.4
11	1	142.0292510	0.5	0.8	0.5300858	0.1	1.3	0.0000000	100.0	0.1
11	2	141.2880250	0.0	1.9	0.5300858	0.1	1.3	0.0014825	0.0	2.0
11	3	141.2880250	0.0	1.9	0.5292941	0.0	2.3	0.0014825	0.0	1.9
11	4	141.2880710	0.0	2.6	0.5292941	0.0	2.4	0.0014820	0.0	3.0
11	5	141.2880710	0.0	2.6	0.5292966	0.0	4.4	0.0014820	0.0	3.0

Table 19 f(x)=vd(x), step h=10

Interpolation points:

j	x_j	y_j
-5	995.00000	138.66008
-4	996.00000	139.18274
-3	997.00000	139.70685
-2	998.00000	140.23247
-1	999.00000	140.75954
0	1000.00000	141.28809
1	1001.00000	141.81815
2	1002.00000	142.34962
3	1003.00000	142.88266
4	1004.00000	143.41711
5	1005.00000	143.95314

Function value : 141.2880860
First derivative : 0.5292947
Second derivative : 0.0014820

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	141.2885900	0.0	0.5	0.5293045	0.0	0.6	0.0000000	100.0	0.2
3	2	141.2880860	0.0	0.8	0.5293045	0.0	0.7	0.0015299	3.2	0.9
5	1	141.2895810	0.0	0.5	0.5292923	0.0	0.8	0.0000000	100.0	0.1
5	2	141.2881010	0.0	1.2	0.5292923	0.0	0.8	0.0014757	0.4	1.2
5	3	141.2881010	0.0	1.2	0.5293096	0.0	1.3	0.0014757	0.4	1.3
5	4	141.2880860	0.0	1.2	0.5293096	0.0	1.3	0.0015597	5.2	1.7
7	1	141.2910610	0.0	0.6	0.5292985	0.0	1.0	0.0000000	100.0	0.2
7	2	141.2881010	0.0	1.4	0.5292985	0.0	1.0	0.0014798	0.1	1.4
7	3	141.2881010	0.0	1.4	0.5292897	0.0	1.8	0.0014798	0.1	1.4
7	4	141.2881010	0.0	2.1	0.5292897	0.0	1.8	0.0014687	0.9	2.4
7	5	141.2881010	0.0	2.0	0.5293134	0.0	3.2	0.0014687	0.9	2.4
11	1	141.2955020	0.0	0.7	0.5293015	0.0	1.3	0.0000000	100.0	0.1
11	2	141.2881010	0.0	1.9	0.5293015	0.0	1.3	0.0014803	0.1	2.0
11	3	141.2881010	0.0	1.9	0.5292926	0.0	2.3	0.0014803	0.1	2.0
11	4	141.2881010	0.0	2.6	0.5292926	0.0	2.4	0.0014738	0.6	3.0
11	5	141.2881010	0.0	2.6	0.5293013	0.0	4.3	0.0014738	0.6	3.0

Table 20 f(x)=vd(x), step h=1

Interpolation points:

j	x_j	y_j
-5	999.50000	141.02361
-4	999.59998	141.07648
-3	999.70001	141.12936
-2	999.79999	141.18225
-1	999.90002	141.23518
0	1000.00000	141.28809
1	1000.09998	141.34103
2	1000.20001	141.39398
3	1000.29999	141.44690
4	1000.40002	141.49995
5	1000.50000	141.55290

Function value	:	141.2880860
First derivative	:	0.5292947
Second derivative	:	0.0014820

Nb pts	Deg	Estimated function	Error %	CPU 1000	Estimated derivative	Error %	CPU 1000	Estimated 2nd derivative	Error %	CPU 1000
3	1	141.2881010	0.0	0.5	0.5292511	0.0	0.7	0.0000000	100.0	0.1
3	2	141.2880860	0.0	0.8	0.5292511	0.0	0.6	0.0065104	339.3	0.9
5	1	141.2881160	0.0	0.5	0.5293121	0.0	0.8	0.0000000	100.0	0.1
5	2	141.2881160	0.0	1.2	0.5293121	0.0	0.8	0.0006200	58.2	1.3
5	3	141.2881160	0.0	1.2	0.5292256	0.0	1.4	0.0006200	58.2	1.3
5	4	141.2880860	0.0	1.2	0.5292256	0.0	1.3	0.0063382	327.7	1.7
7	1	141.2881160	0.0	0.6	0.5292565	0.0	1.0	0.0000000	100.0	0.1
7	2	141.2881010	0.0	1.4	0.5292565	0.0	1.0	0.0006200	58.2	1.4
7	3	141.2881010	0.0	1.4	0.5293455	0.0	1.8	0.0006200	58.2	1.4
7	4	141.2881010	0.0	2.0	0.5293455	0.0	1.8	0.0003207	78.4	3.2
7	5	141.2881010	0.0	2.3	0.5292028	0.0	3.4	0.0003207	78.4	2.3
11	1	141.2881470	0.0	0.7	0.5293010	0.0	1.4	0.0000000	100.0	0.1
11	2	141.2880860	0.0	1.9	0.5293010	0.0	1.3	0.0012141	18.1	2.1
11	3	141.2880860	0.0	1.9	0.5292820	0.0	2.4	0.0012141	18.1	2.1
11	4	141.2881010	0.0	2.6	0.5292820	0.0	2.3	0.0014659	1.1	3.1
11	5	141.2881010	0.0	3.1	0.5292293	0.0	4.8	0.0014659	1.1	3.5

Table 21 f(x)=vd(x), step h=0.1

From these results, one may draw the following conclusions:

a) Whatever the choices of m and n, the precision of the numerical approximation depends to a large extent on the size of the step h. Too large a step results in relatively large errors for all the estimates, while too small a step influences the goodness of the second derivative estimates. The choice of the step size h is discussed further below.

b) The computational times appear to be independent of the functional form of f, which was expected; however it seems to increase linearly with N, for the same value of m, and it increases more significantly as a function of m.

c) The gain in precision obtained for large values of N ($\mathrm{N} = 11$) is not significant. On the other hand the choice of $N = 3$ induces intolerable numerical instability.

For the computation of the first derivative, the two best overall choices, that are suggested by these tests, are:

$$N = 5 \qquad (n = 2) \qquad m = 3$$

and

$$N = 7 \qquad (n = 3) \qquad m = 5 \quad .$$

The first choice requires four function evaluations and the second six function evaluations (as the point $y_0 = f(x_0)$ is not used in the approximation formulae). It is clear that when the evaluation of the function is costly, the first choice is preferable. We use this choice in the following analyses.

9.4 THE APPROXIMATION FORMULA FOR THE DERIVATIVE

From (1) one obtains :

$$f'(x_0) = \frac{1}{h}\sum_{i=-2}^{2}\sum_{k=0}^{3}\frac{P_{k,2}(i)P'_{k,2}(0)}{\sum_{j=-2}^{2}[P_{k,2}(j)]^2}\,y_i$$

But (Hildebrand (1974), p.351):

$P_{0,2}(x) = 1$

$P_{1,2}(x) = x/2$

$P_{2,2}(x) = (x^2 - 2)/2$

$P_{3,2}(x) = (5x^3 - 17x)/6$

By substituting in the formula for the derivative, one obtains

$$f'(x_0) = \tfrac{1}{12h}(y_{-2} - y_2 - 8(y_{-1} - y_1)) \qquad (2)$$

9.5 CHOICE OF THE STEP h

In a procedure that computes the derivative, the choice of the step h should be made automatically. It is necessary to make a "good" choice, since the precision of the approximation depends largely on this choice.

For the four test functions used previously, we plotted the relative error made on the approximation of the derivative as a function of $log_{10}(h)$ for different values of x (see Figures 6 to 15). A "confidence" interval, that is, the interval where the relative error is less than 0.1% is given in each case.

The initial choice of h may be based only on data known where the procedure is initiated, which are x_0 and $y_0 = f(x_0)$. Figures 16 and 17 give the best choice for the step h, as well as the confidence interval of 0.1%, as a function of these choices.

It is not surprising that there is no particular discernable pattern in the graph of h vs y_0, as the derivative is not influenced by the addition of a constant term to the function f. On the other hand one can detect a rough relationship between h and x_0. In Figure 18, a "linear confidence region" was traced manually and the line in the middle provides the relation that is used to determine the step size:

$$\log_{10}(h_0) = 0.95\log_{10}(x_0) - 2$$

hence

$$h_0 = 10^{(0.95\log_{10}(x_0)-2)} \tag{3}$$

Once the initial step size h_0 is chosen, $f(x_0 - h_0)$ and $f(x_0 + h_0)$ may be computed in order to obtain a first approximation of the derivative at x_0:

$$f'(x_0) = \frac{f(x_0 + h_0) - f(x_0 - h_0)}{2h_0} = Z$$

By using Z, we may refine the initial step chosen as follows:

$$\begin{array}{ll} \text{if Z} \leq 1 \text{ then} & \text{h} := h_0 \\ \text{otherwise} & \text{h} := h_0/2 \end{array}$$

In the first case, the two new interpolation points are computed at $(x_0 + 2h)$ and $(x_0 - 2h)$ and in the second case at $(x_0 + h/2)$ and $(x_0 - h/2)$.

9.6 SOME FINAL TESTS

We applied the approximation procedure which uses formula (2) and the initial step h_0 to the four functions used in this analysis. In 74 tests, only in 2 cases the relative error was higher than 0.1% and in 71 cases the relative error was less than 0.1%. The three "bad" results occured for the function TG in the neighbourhood of $\frac{\pi}{2}$. For each of these tests, we compared the approximations obtained with those obtained with a polynomial of degree 5 with 7 interpolation points. These results were better than those obtained with the polynomial of degree 3 only in 24% of the cases.

From the point of view of computational efficiency we noted that most of the time is used to compute h_0 (log + exponentiate). The formula for h_0 may simplified to:

$$h_0 = 10^{1.log_{10}(x_0)-2} = \frac{x_0}{100}$$

with significant savings in computational time and little effect on the quality of the approximations (Guélat (1987)).

9.7 PERFORMANCE FOR STAN FUNCTIONS

The STAN assignment procedure (see Chapter 3 for details) requires the evaluation of marginal transportation costs:

$$C_{\bar{e}}^{\bar{p}} = s_{\bar{e}}^{\bar{p}} + \sum_{p} \sum_{e} \frac{\partial s_{e}^{p}}{\partial v_{\bar{e}}^{\bar{p}}} v_{e}^{p}$$

where $\bar{p}$ represents a commodity, $\bar{e}$ a link of the network, s_{e}^{p} the cost function specified by the user and v_{e}^{p} represents the volume of product p on link e.

The large number of computations of partial derivatives required by these functions motivated our search for efficient evaluation procedures (for example, the exact calculation of the marginal cost on function F7 required an analytical development which took about 18 handwritten pages!). Table 22 compares the precision of the approximation and exact estimation procedures for functions of increasing complexity (see Part III for the definition of variables used). The results are very conclusive and the approximation method is used in the STAN assignment procedure.

mean cost	marginal cost			relative
function	exacted	/	estimated	error %
F1 = $length$	59.00	/	59.00	0.00
F2 = $length * (1 + voltot)$	659.00	/	659.00	0.00
F3 = $length * (1 + strain)$	202.10	/	202.10	0.00
F4 = $length * (1 + vehprod)^{1.5}$	234.50	/	234.50	0.00
F5 = $length * (1 + vehtot)^{1.5}$	601.60	/	601.60	0.00
F6 = $length * (1 + e^{train/5})$	249.50	/	250.00	0.06
F7 = $length * (1 + \frac{train}{strain+1} +$ $+ \frac{vehprod}{vehtot+1} + \frac{volprod}{voltot+1})$	70.73	/	71.41	0.96

Table 22 Estimation of Marginal Costs

9.8 CONCLUSIONS

Formula 2 offers, despites its simplicity, a good approximation of the derivative of f at x_0 and the use of a higher order formulae is not required. The chosen step, h_0, has also efficiently performed and is certainly adequate for monotone functions. These approximations should be used with caution in the neighbourhood of one of the "zeroes" of the function.

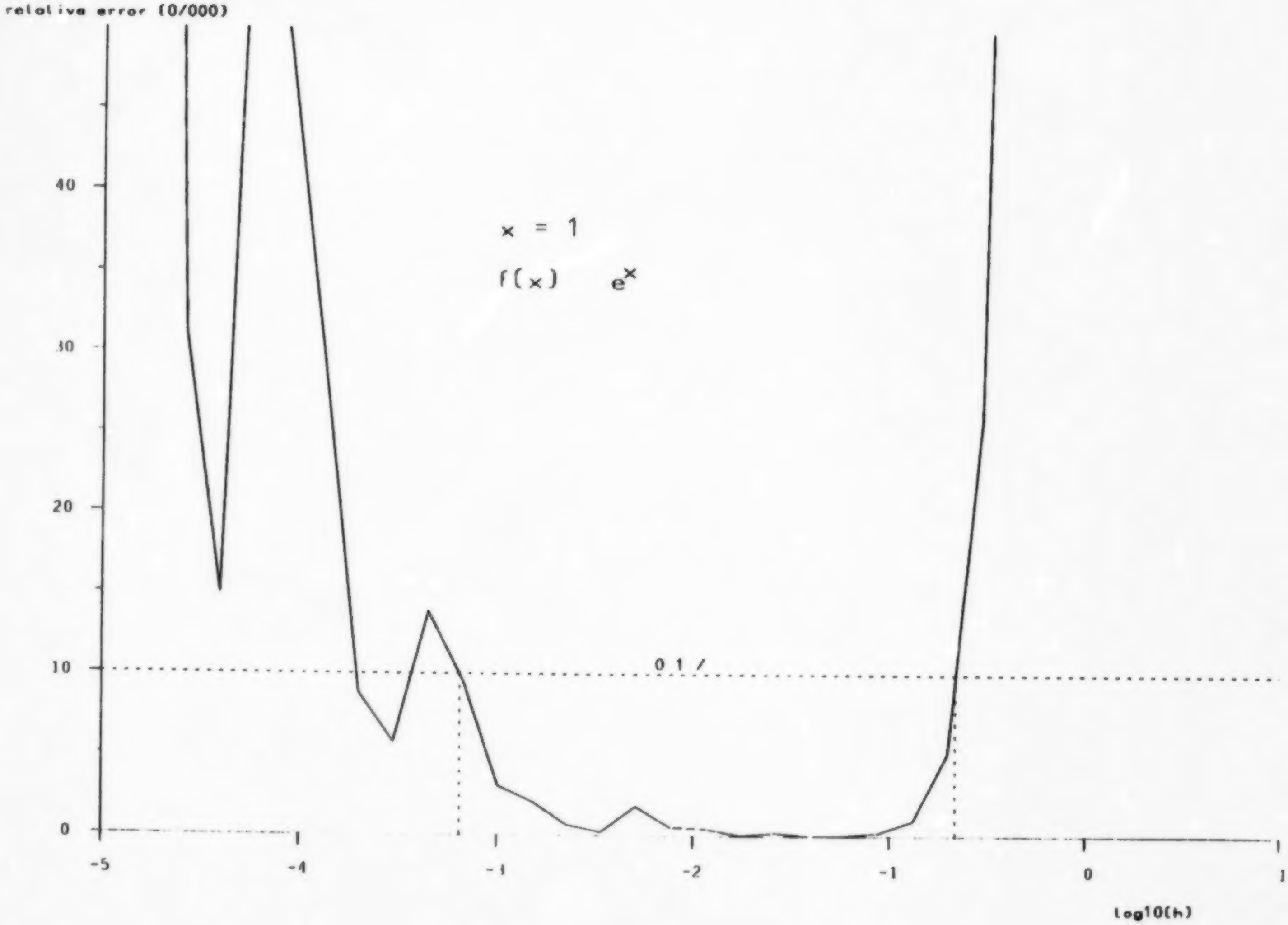


Figure 6: EXP, x=1

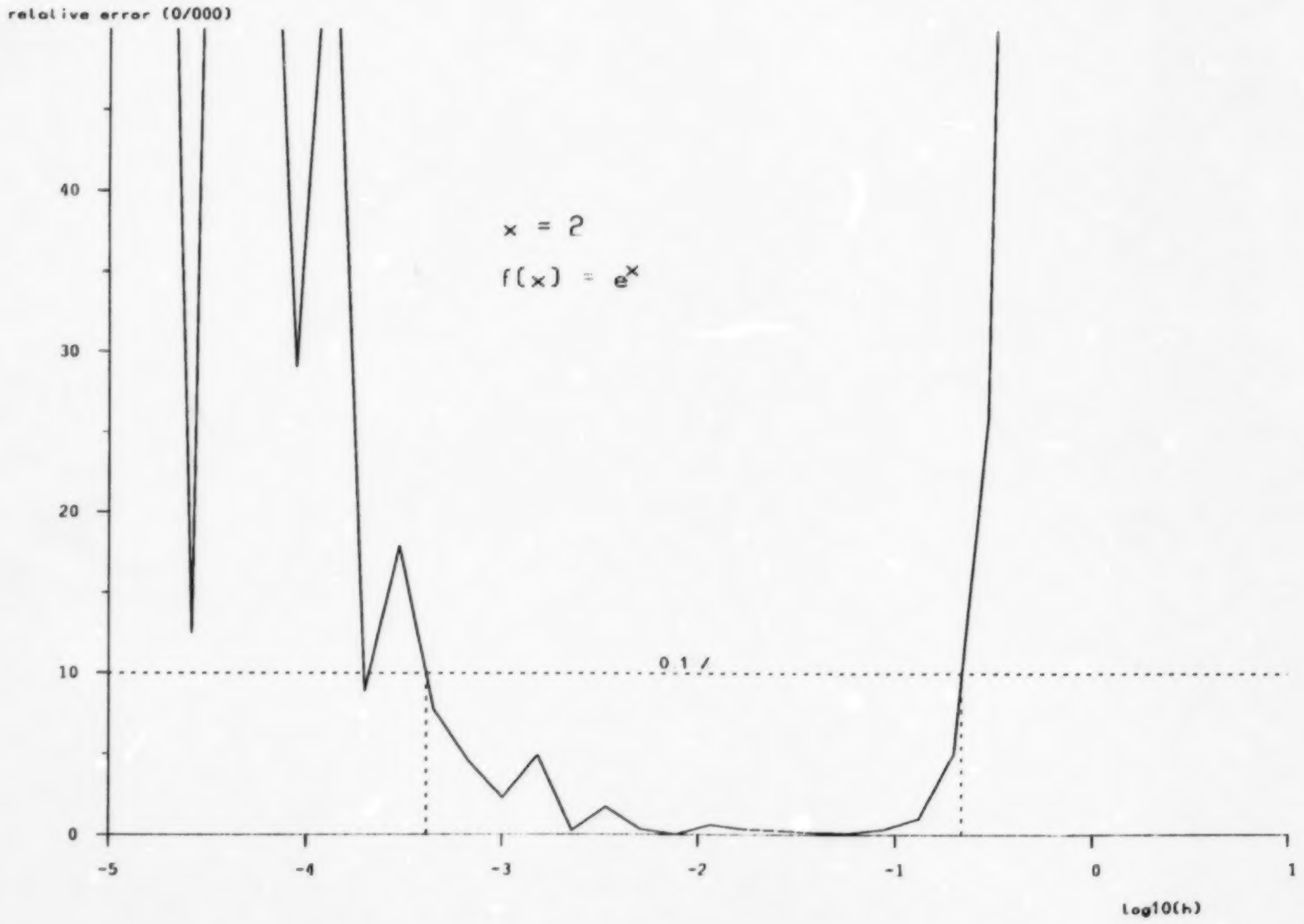


Figure 7: EXP. x=2

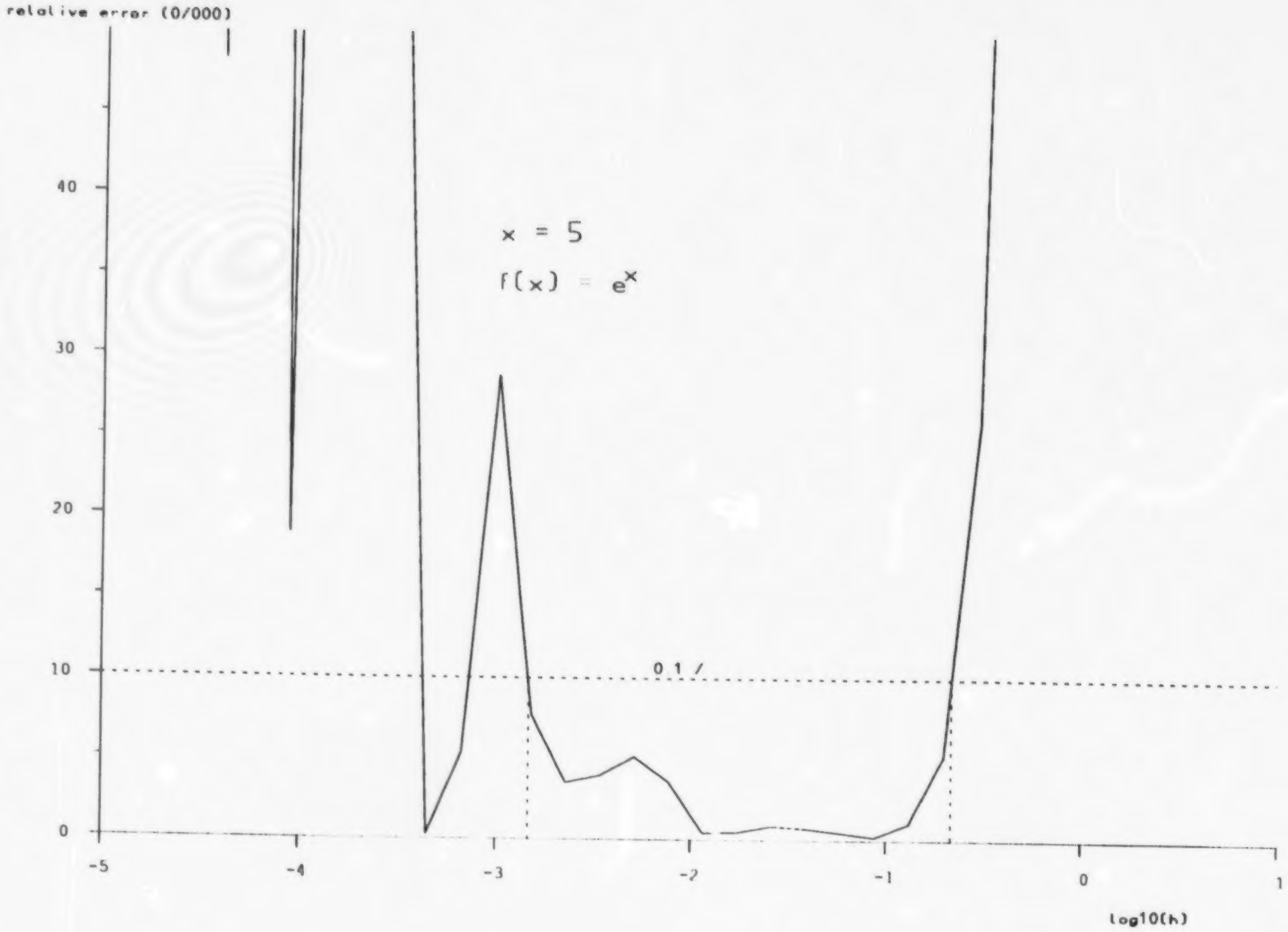


Figure 8: EXP. x=5

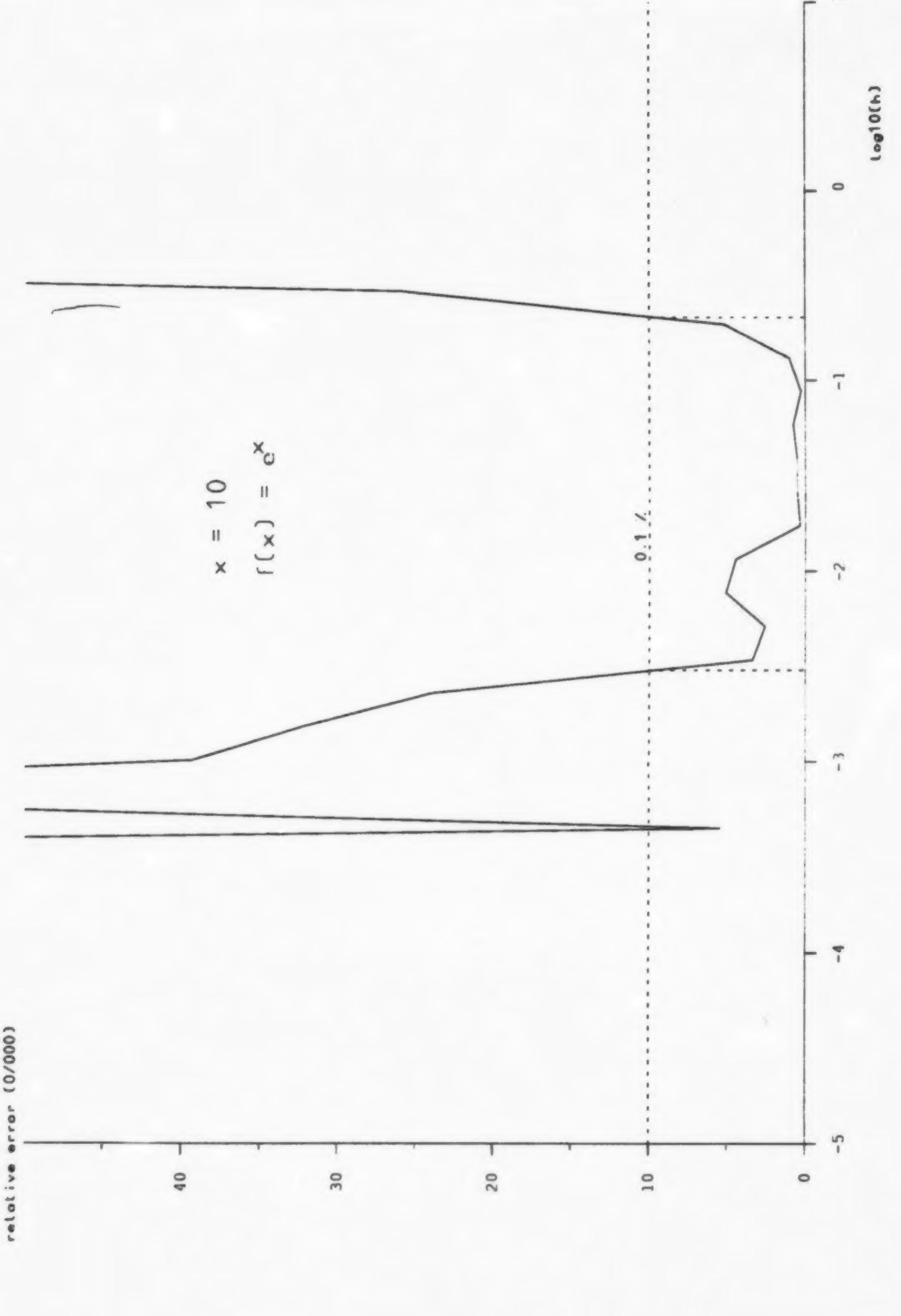


Figure 9: EXP, x=10

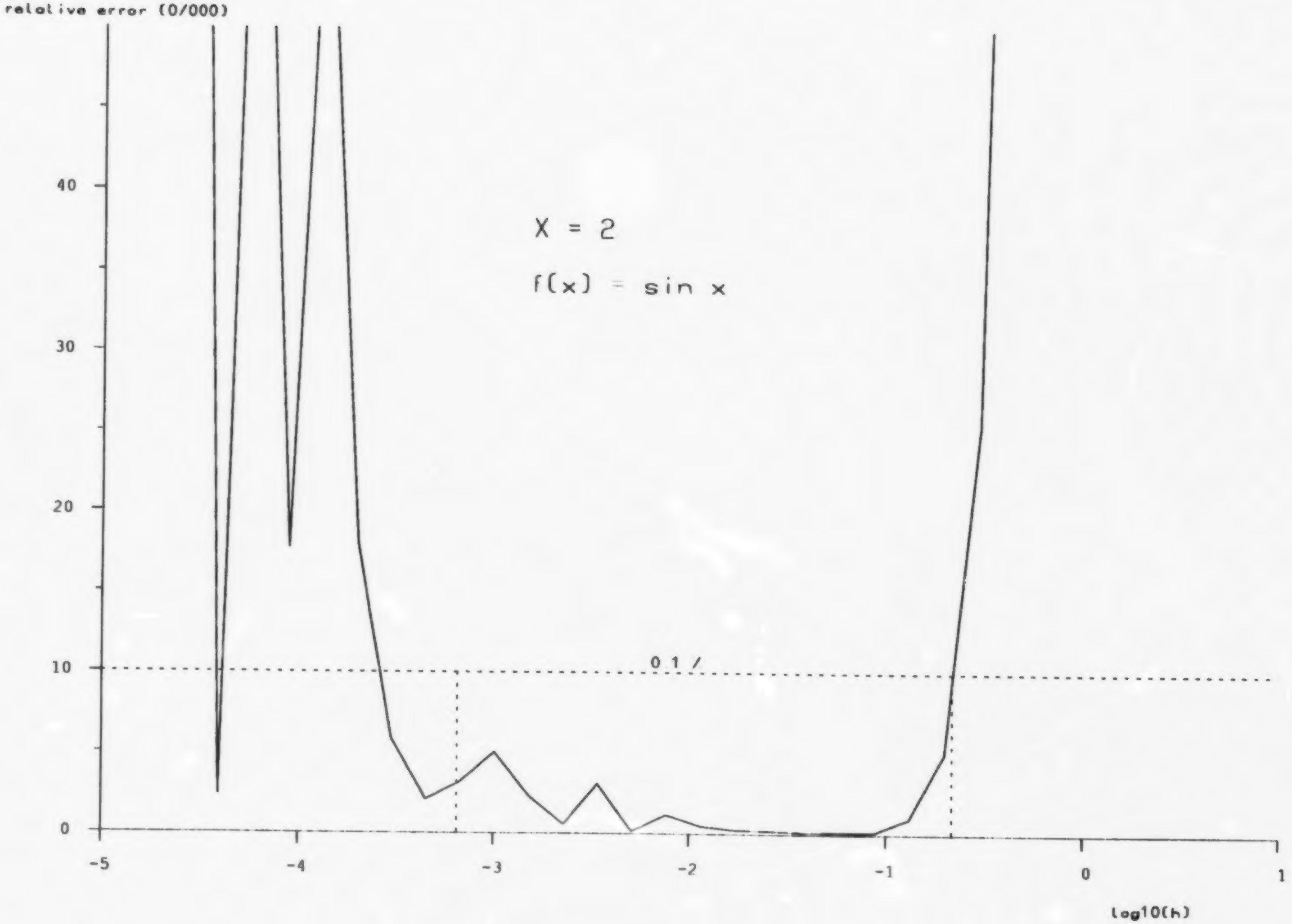
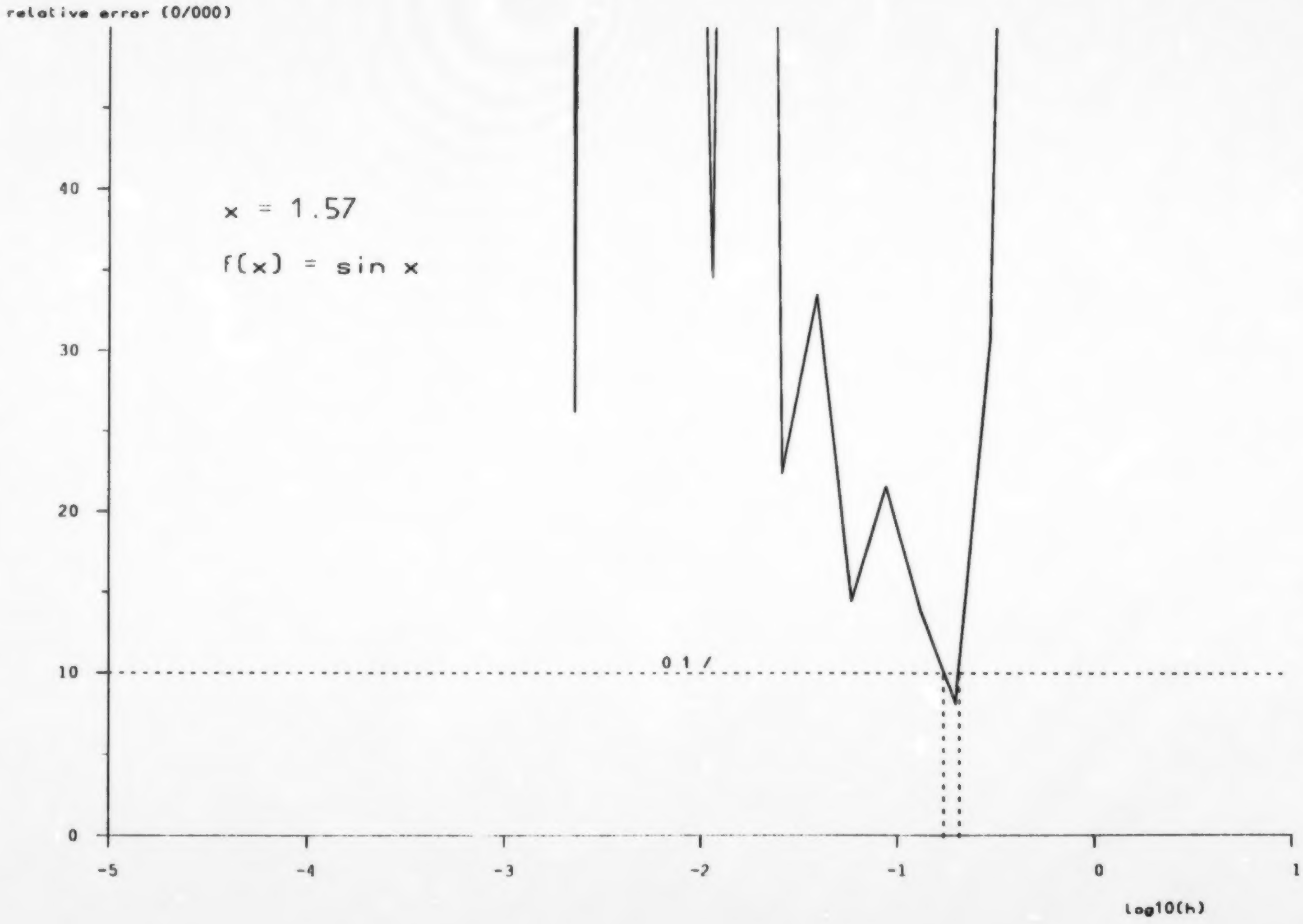


Figure 10: SIN, x=2



igure 11: SIN. x=1.57

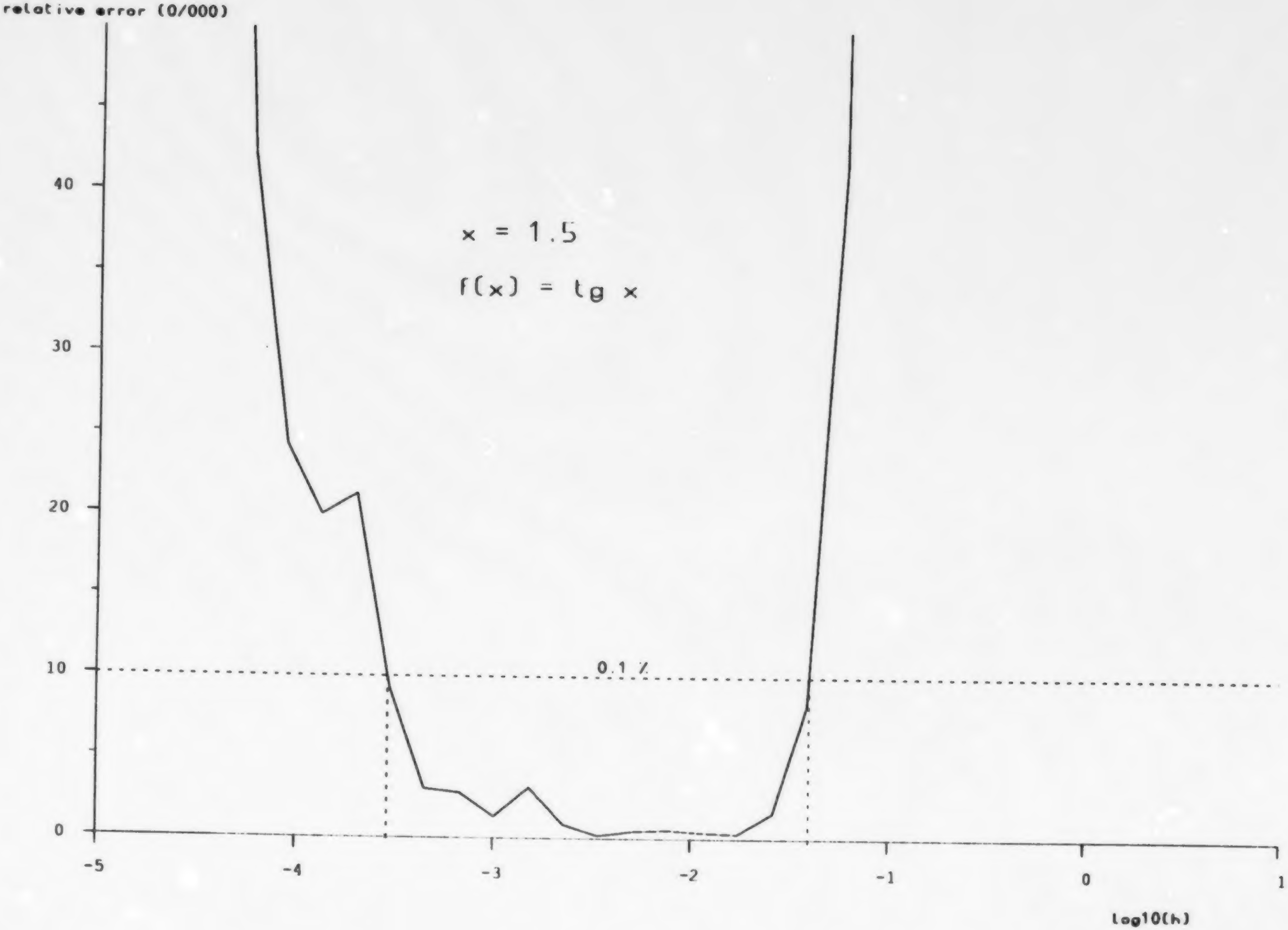


Figure 12: IG. x=1.5

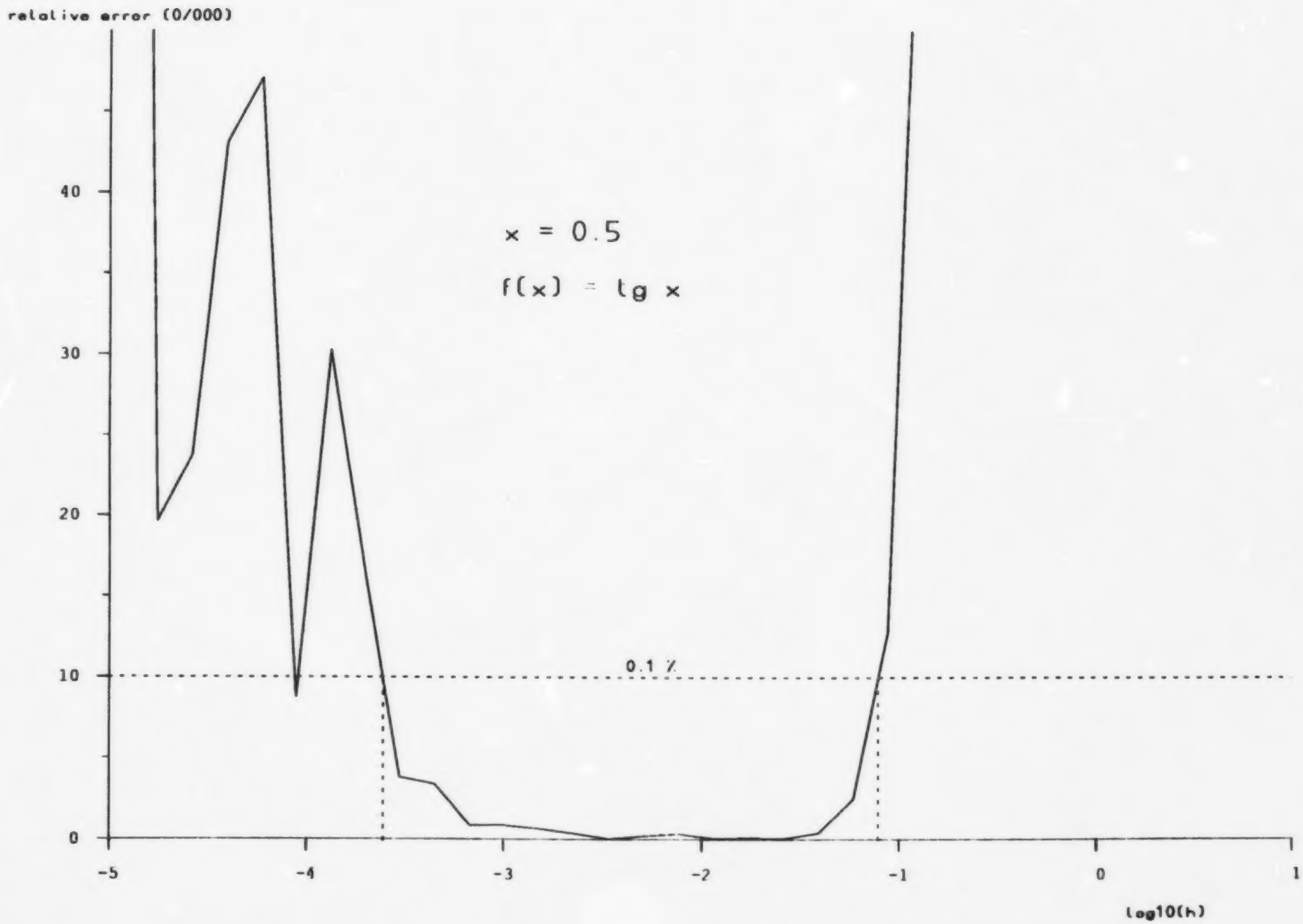


Figure 13: TG. x=0.5

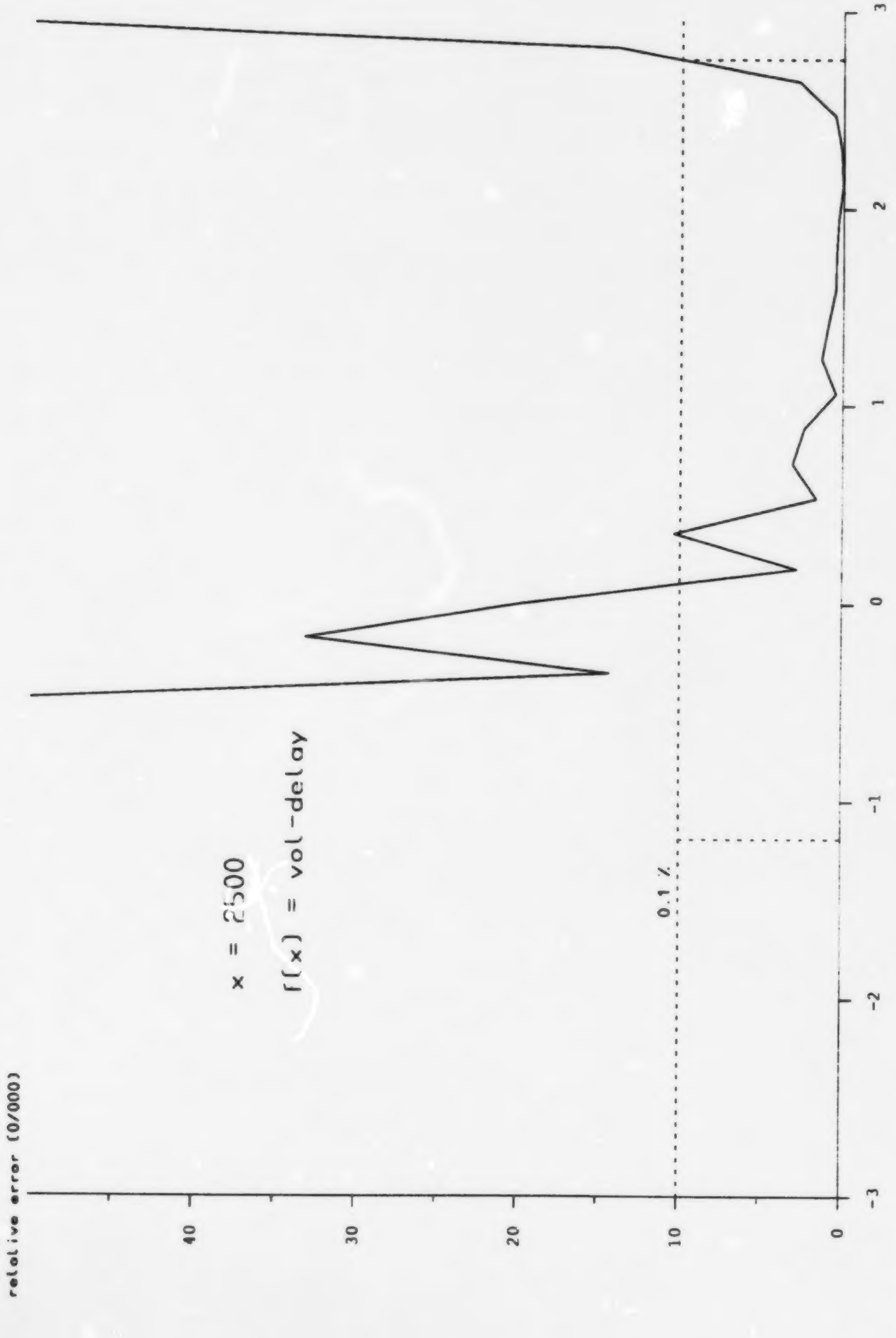


Figure 14: VD. x=2500

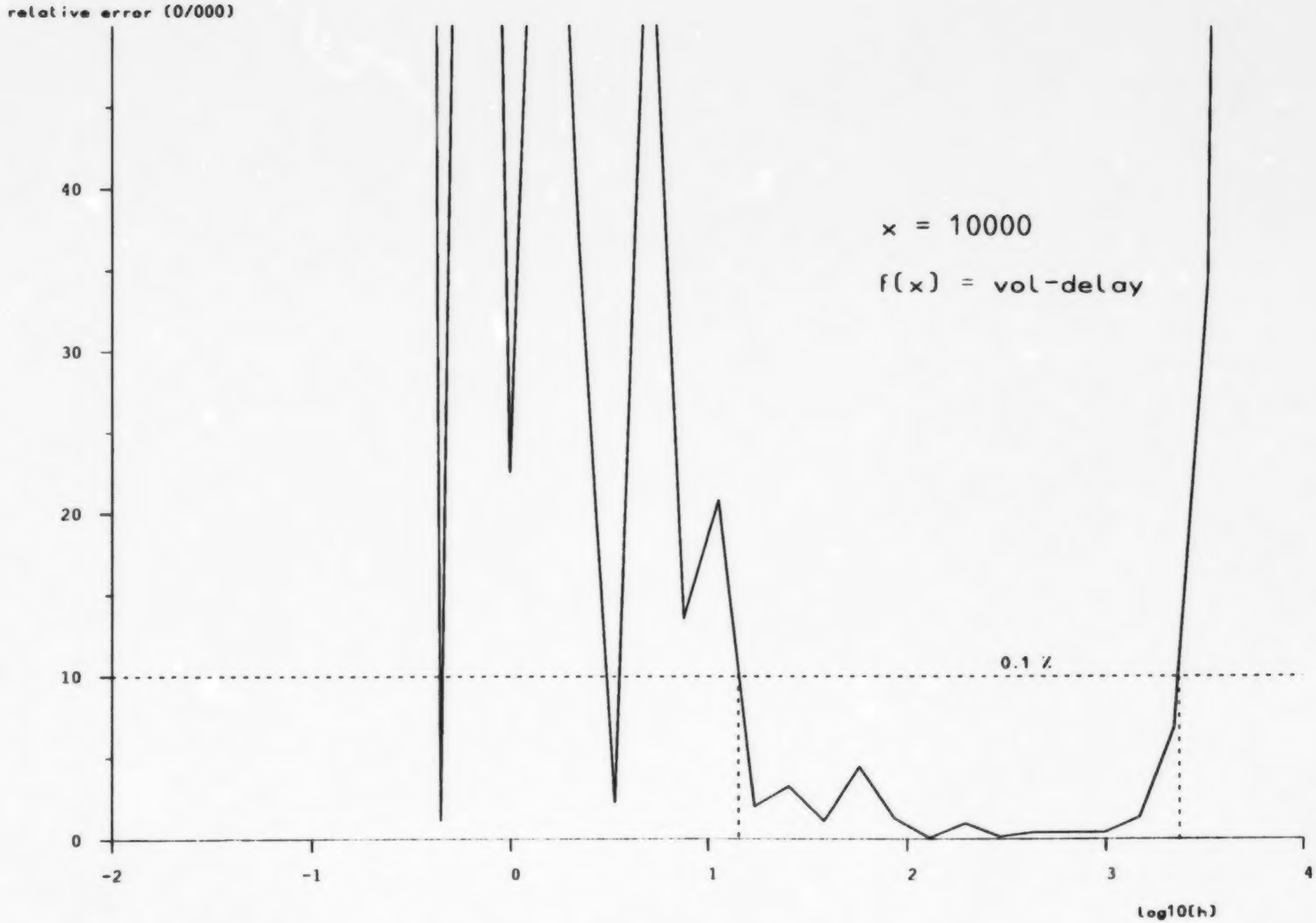


Figure 15: VD. x=10000

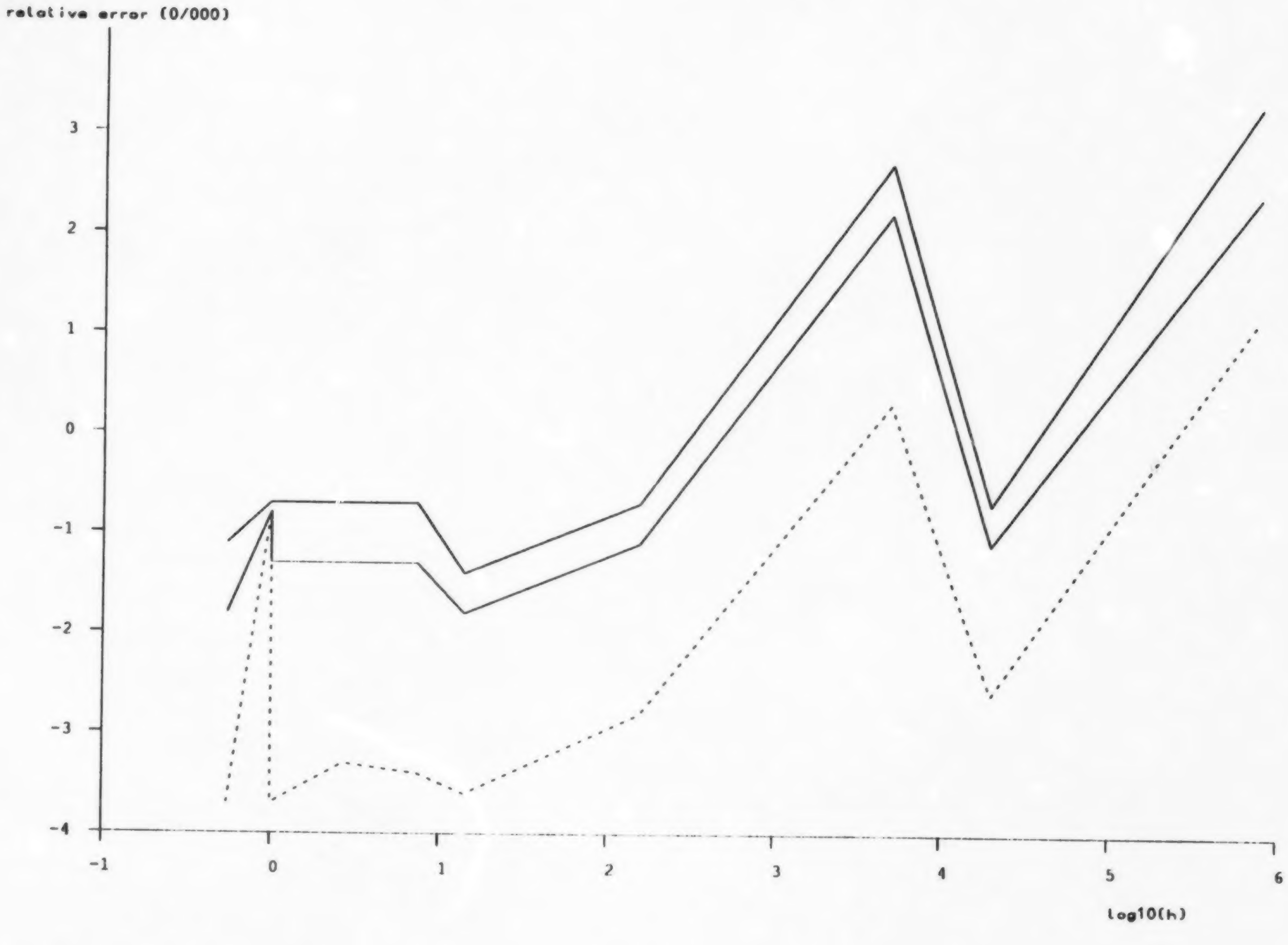


Figure 16: optimal step h vs y_0

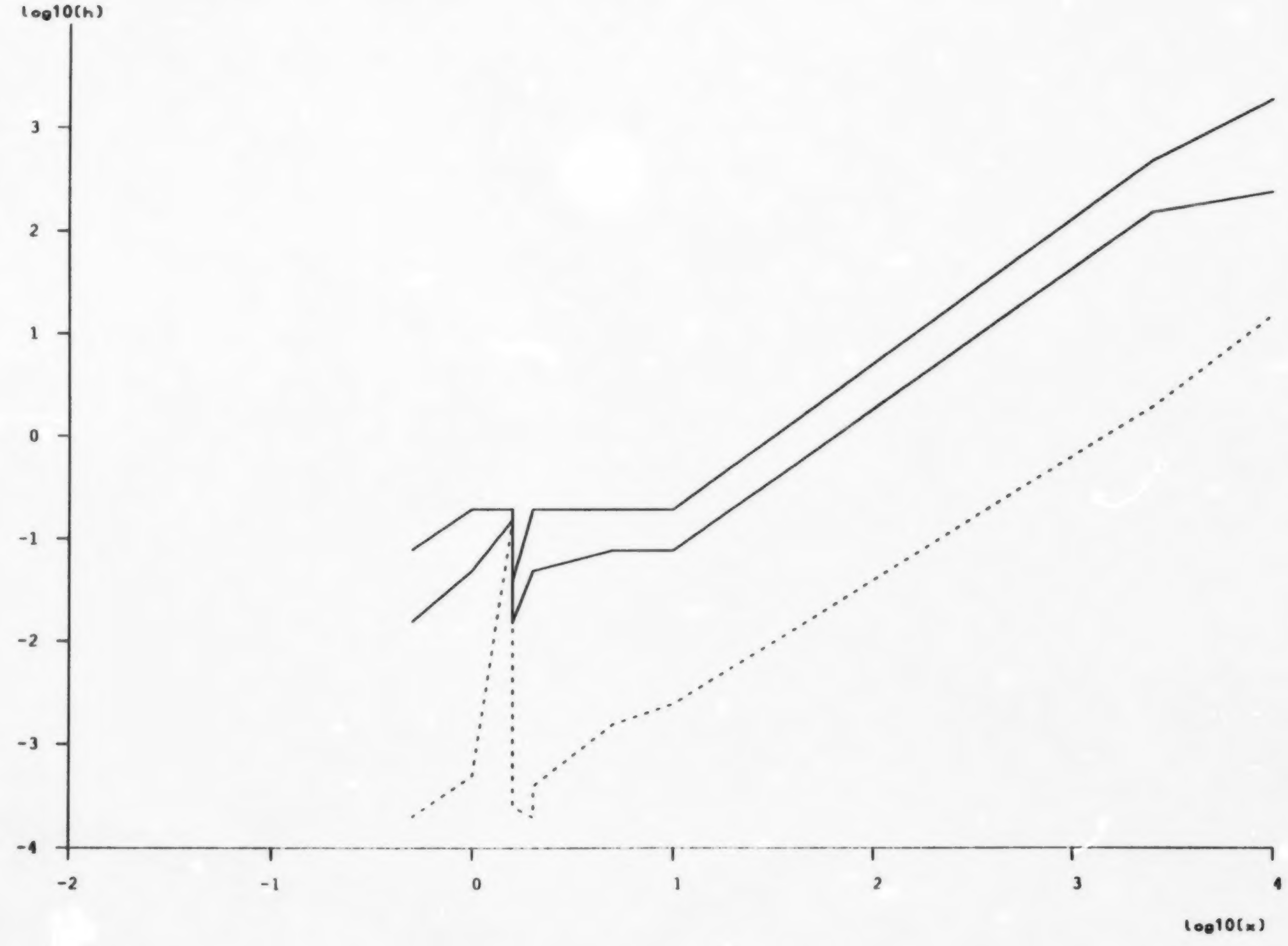


Figure 17: optimal step h vs x_0

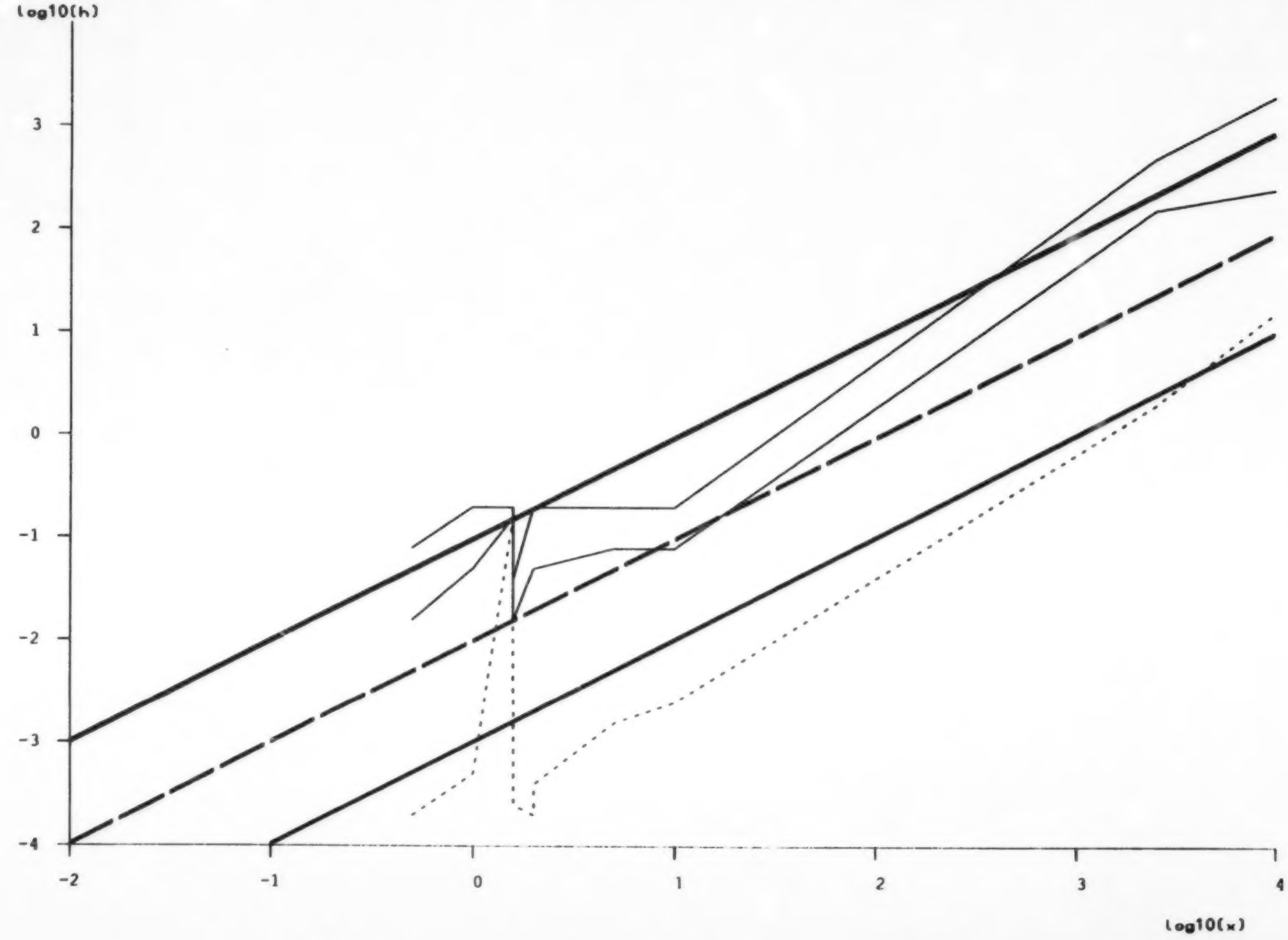


Figure 18: optimal step h_0 selected

9.9 BIBLIOGRAPHY

Casaletto, J.J. and Rice, J.R. (1978), *Smoothing and Estimating Derivatives of Equispaced Data*, **Applicable Analysis** vol. 8,143-152.

Guélat, J. (1987), *Calcul des dérivées (partielles) des fonctions de coût dans le progiciel STAN*, publication #535, Centre de recherche sur les transports, Université de Montréal.

Hildebrand, F.B. (1974), *Introduction to Numerical Analysis*, McGraw-Hill Inc. (2nd edition).

Kellison, S.G. (1975), *Fundamentals of Numerical Analysis*, Richard D. Irwin Ink., Homewood, Illinois.